NASA Contractor Report 3442

A Computer Code for Swirling Turbulent Axisymmetric Recirculating Flows in Practical Isothermal Combustor Geometries

D. G. Lilley and D. L. Rhode

GRANT NAG 3-74
FEBRUARY 1982

NASA

NASA Contractor Report 3442

A Computer Code for Swirling Turbulent Axisymmetric Recirculating Flows in Practical Isothermal Combustor Geometries

D. G. Lilley and D. L. Rhode
Oklahoma State University
Stillwater, Oklahoma

Prepared for
Lewis Research Center
under Grant NAG 3-74

NASA
National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1982

CONTENTS Page

1. INTRODUCTION

1.1 The Problem

In combustion chamber development, designers are aided by experiments but, as a supplement to them, economical design and operation can be greatly facilitated by the availability of prior predictions of the flowfield. These may be obtained by use of a mathematical model incorporating a numerical finite difference prediction procedure. This work combines the rapidly developing fields of theoretical combustion aerodynamics and computational fluid dynamics, and its improvement and use will significantly increase understanding and reduce the time and cost of development.[1-6]

The present research work is concerned with complementary experimental and theoretical studies and is described in a recent paper.[7] The problem being investigated is restricted to steady turbulent flow in axisymmetric geometries, under low speed and nonreacting conditions - a study area highlighted recently[8-10] as a fundamental research requirement in combustion modeling. The particular problem is concerned with turbulent flow of a given turbulence distribution in a round pipe entering an expansion into another round pipe, as illustrated in Fig. 1. The in-coming flow may possess a swirl component of velocity via passage through swirl vanes at angle ϕ [equal approximately to $\tan^{-1}(w_{in}/u_{in})$], and the side-wall may slope at an angle α, to the main flow direction. The resulting flowfield domain may possess a central toroidal recirculation zone CTRZ in the middle of the region on the axis, in addition to the possibility of a corner recirculation zone CRZ near the upper corner provoked by the rather sudden enlargement of the cross-sectional area. Of vital concern is the characterization of flows of this type in terms of the <u>effects</u> of side-wall angle α, degree of swirl ϕ, turbulence intensity k_{in} of the inlet stream and expansion ratio D/d <u>on</u> the resulting flowfield in terms of its time-mean and turbulence quantities.

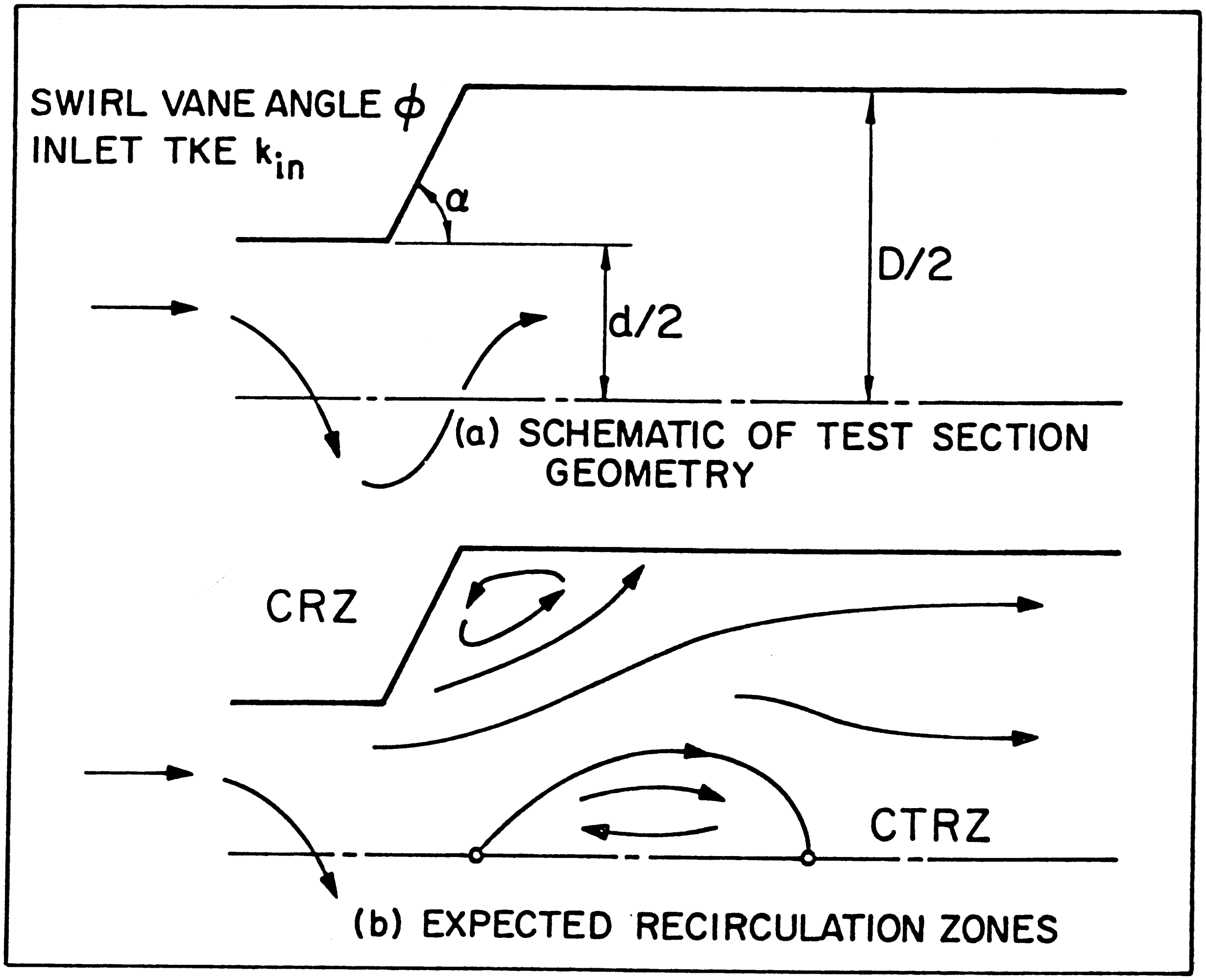


Fig. 1. The flowfield being investigated.

Such problems have received little attention, yet there is a definite need for work in this area even under nonreacting flow conditions.

The objective of the present document is to present details of the computational solution procedure which has been developed to perform the prediction function in the configuration just described. The finally developed computer program (written in Fortran 4) is code-named STARPIC (mnemonic for swirling turbulent axisymmetric recirculating flows in practical isothermal combustor geometries) and it is the purpose of this document to serve as a user's manual and describe it in sufficient detail for it to be used and amended by persons whose interest is confined more to the basis and results of the computations.

1.2 Previous Work

A mathematical solution of the flowfield of interest should provide results, if possible more cheaply, quickly, and correctly than possible by other means (for example, experiments on real-life systems or models). In order to achieve this, the model should simulate the flow in all its important respects (geometry, boundary conditions, physical properties of gases, turbulence, etc.) and provide a means whereby the governing equations may be solved. Mathematical models of steadily increasing realism and refinement are now being developed, both in the dimensionality of the model (together with the computational procedures) and in problems associated with the simulation of the physical processes occurring. Clearly there are two areas of difficulty: the simulation and solution.

Several previous publications discuss these problems at length, in terms of practical application,[5,6] turbulence modeling[11,12] and numerical solution of 2-D axisymmetric problems via the stream function ~ vorticity or primitive pressure ~ velocity approach.[13-15] Whereas the former approach, used in the 1968 computer program from Imperial College for example[16], reduces by one the number of equations to be solved and eliminates the troublesome pressure (at the expense of trouble with the vorticity equation), the preferred approach now is SIMPLE (mnemonic for semi-implicit method for pressure linked equations) which focuses attention directly on the latter variables. Because it possesses many advantages, the present work has been developed immediately on this new technique, the basic ideas of which have been embodied into the 1974 Imperial College TEACH (teaching elliptic axisymmetric characteristics heuristically) computer program.[17] Advances to this code have been made elsewhere and applied to other 2-D[18-25] and 3-D[26-29] problems of interest to the combustor designer.

1.3 Outline of the Present Contribution

The present document serves as a user's manual for the STARPIC computer program, with three subsequent major sections dealing with the basic ideas, the computer program and the user's guide. Consideration is given to recent work in the finite difference solution, via a primitive variable code, of axisymmetric swirl flow in the combustor geometry of Fig. 1, where the inlet expansion sidewall may slope obliquely to the central axis. Section 2 describes the mathematical problem, by providing the basic equations and assumptions for the flow, and discussing the primitive variable solution technique to be used. Section 3 concerns itself with the organization and structure of the computer program. Sub-sections of the program are here described and linked with the equations previously noted in the text; and explanations are given of the special features of the program. Readers primarily interested in using the program, without concerning themselves greatly with the computational details, can give their main attention to Section 4. This is a user's guide to the program and the emphasis is on those parts of the program which will require alteration to suit particular flow configurations, to obtain the desired output, to test for accuracy and to improve it when necessary. Appendices are provided dealing with the hydrid differencing scheme and reasons for its superiority, a Fortran symbol list, a listing of the STARPIC computer program, The program output is included in a microfiche supplement at the back of the report.

2. THE MATHEMATICAL PROBLEM

2.1 The Governing Equations

The turbulent Reynolds equations for conservation of mass, momentum (in x, y and θ directions), turbulence energy k and turbulence dissipation rate ε, which govern the 2-D axisymmetric, swirling, steady flow may be taken as previously.[19-15] The transport equations are all similar and contain terms for convection and diffusion (via turbulent flux terms) and source S_ϕ of a general variable (which contains terms describing the generation (creation) and consumption (dissipation) of ϕ). Introducing turbulent exchange coefficients and the usual turbulent diffusion-flux (stress ~ rate of strain type) laws, it can be shown that the similarity between the differential equations and their diffusion relations allows them all to be put in the common form:

$$\frac{1}{r}\left[\frac{\partial}{\partial x}(\rho u r \phi) + \frac{\partial}{\partial r}(\rho v r \phi) - \frac{\partial}{\partial x}\left(r\Gamma_\phi \frac{\partial \phi}{\partial x}\right) - \frac{\partial}{\partial r}\left(r\Gamma_\phi \frac{\partial \phi}{\partial r}\right)\right] = S_\phi \qquad (1)$$

for $\phi = 1$ (continuity equation), u, v, w (three velocity components), k and ε (two turbulence quantities). The forms for the source term S_ϕ are given in Table 1, where certain quantities are defined as follows:

$$S^u = \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r\mu \frac{\partial v}{\partial x}\right) \qquad (2)$$

$$S^v = \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial r}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r\mu \frac{\partial v}{\partial r}\right) \qquad (3)$$

$$G = \mu\left[2\left\{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial r}\right)^2 + \left(\frac{v}{r}\right)^2\right\} + \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x}\right)^2 + \left\{r\frac{\partial}{\partial r}(w/r)\right\}^2 + \left(\frac{\partial w}{\partial x}\right)^2\right] \qquad (4)$$

Implicit here is the use of the two-equations k-ε turbulence model[11,12]

Table 1. The Form of the Source Term in the General Equation for ϕ (Eq. (1))

ϕ	S_ϕ
u	$-\frac{\partial p}{\partial x} + S^u$
v	$-\frac{\partial p}{\partial r} + \frac{\rho w^2}{r} - \frac{2\mu v}{r^2} + S^v$
w	$-\frac{\rho v w}{r} - \frac{w}{r^2}\ \frac{\partial}{\partial r}\ (r\mu)$
k	$G - C_D \rho \varepsilon$
ε	$(C_1 \varepsilon G - C_2 \rho \varepsilon^2)/k$

$$\mu \;=\; C_\mu \rho k^2/\varepsilon \;+\; \mu_\ell \tag{5}$$

$$\Gamma_\phi = \mu/\sigma_\phi \tag{6}$$

which is commonly used in computer codes for turbulent flow prediction.

These equations have to be solved for the time-mean pressure p and velocity components u, v, and w. Then other useful designer information like streamline plots showing breakaway and reattachment points, recirculation zones and stagnation points are easily produced. Streamline plots are obtained from the dimensionless stream function which is given by

$$\psi^* = \int_0^r urdr \,/\, \int_0^{d/2} urdr \tag{7}$$

This quantity is calculated for all u-cells, and the coordinates of points constituting each of eleven streamlines (ψ^* = 0.0 through ψ^* = 1.0) is stored for plotting. Earlier publications[17-21] provide details of the present simulation and solution technique, and only highlights of the primitive-variable approach need be given here in the context of the specific problem being investigated.

All the linkages provide a high degree of non-linearity in the total problem, and given the numerical analysis of fluid flow its peculiar difficulty and flavor. The above equations do not alone specify the problem; additional information of two kinds is needed: initial and boundary conditions for all the dependent variables. Details of this are given in Section 2.3. The difficulties in solution spring mainly from the interlinkages between the ϕ's. Those between the axial and radial velocity components are of a peculiar kind, each containing an unknown pressure gradient and the components are linked additionally by another equation, that of mass conservation, in which pressure does not appear. A successful solution procedure is one which takes account of these interactions and ensures that successive adjustments, which must be made to one variable after another, form an involuntary convergent sequence.

2.2 The Solution Technique and Finite Difference Formulation

Solution may be via the stream function-vorticity or primitive pressure-velocity approach, and, as discussed in Section 1.2, the present solution technique is based on the latter. The basic TEACH computer program using SIMPLE with TDMA provides the starting point upon which the present work is based.[17] The following features are incorporated into the program:

(i) a finite difference procedure is used in which the dependent variables are the velocity components and pressure;

(ii) the pressure is deduced from an equation which is obtained by the combination of the continuity equation and the momenta equations (yielding a new form of what is known in the literature as the Poisson equation for pressure);

(iii) the idea is present at each iteration of a first approximation of u, v, and p followed by a succeeding correction;

(iv) the procedure incorporates displaced grids for the axial and radial velocities u and v, which are placed between the nodes where pressure p and other variables are stored; and

(v) an implicity line-by-line relaxation technique is employed in the solution procedure (requiring a tridiagonal matrix to be inverted in order to update a variable at all points along a column), using the TDMA (tri-diagonal matrix algorithm).

The incorporation of these enhances the accuracy and rapidity of convergence of the finally developed computer program.

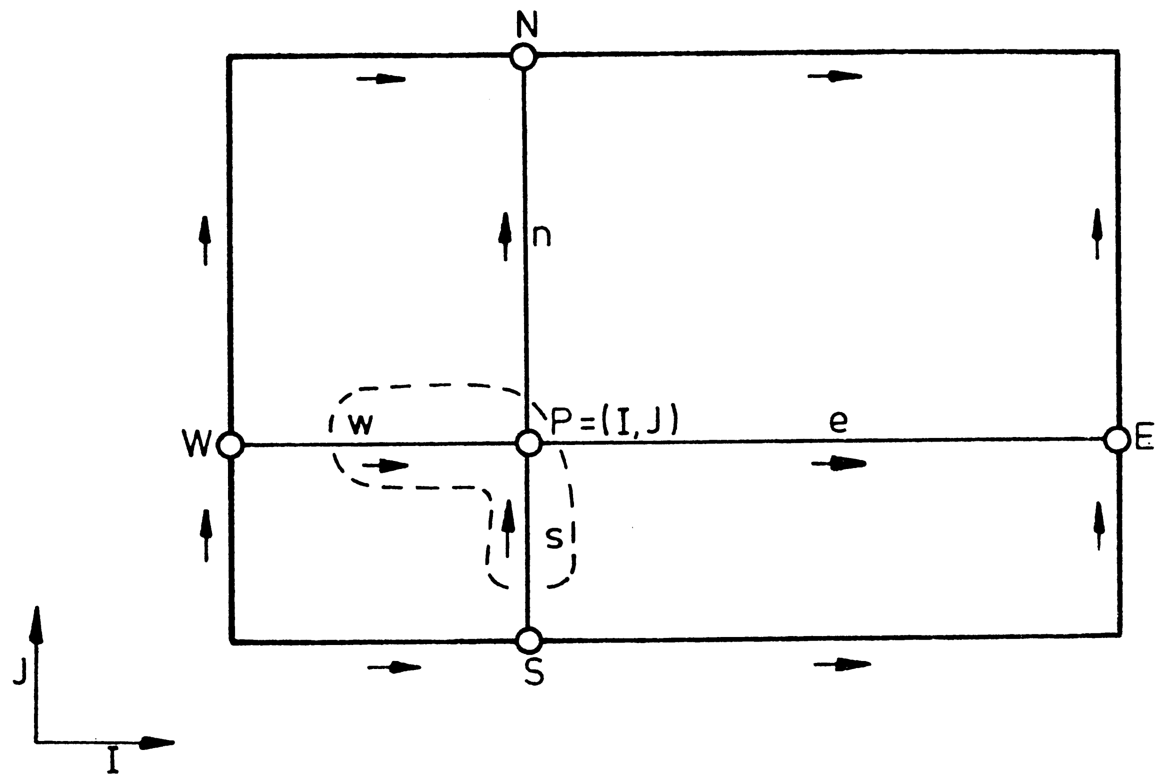
The finite difference equations are solved on a complex mesh illustrated in Fig. 2. The intersections, the point P for example, of the solid lines mark the grid nodes where all variables except the u and v velocity components are stored. The latter are stored at points which are denoted by arrows (and labeled w and s respectively) located midway between the grid intersections, and the boomerang-shaped envelope encloses a triad of points with reference location P at (I,J). Details of the special merits of this staggered grid system have been reported previously.[17,18] The different control volumes C, U and V which are appropriate for the P, w and s locations respectively are given in Fig. 3.

The finite difference equations for each ϕ are obtained by integrating Eq. (1) over the appropriate control volume (centered about the location of ϕ) and expressing the result in terms of neighboring grid point values. The convection and diffusion terms become surface integrals of the convection and diffusive fluxes while the source term is linearized, resulting in

$$
\begin{aligned}
&[\rho u\phi - \Gamma_\phi \frac{\partial\phi}{\partial x}]_e A_e - [\rho u\phi - \Gamma_\phi \frac{\partial\phi}{\partial x}]_w A_w \\
&+ [\rho v\phi - \Gamma_\phi \frac{\partial\phi}{\partial r}]_n A_n - [\rho v\phi - \Gamma_\phi \frac{\partial\phi}{\partial r}]_s A_s = [S_P\phi_P + S_U]\mathrm{Vol} \qquad (8)
\end{aligned}
$$

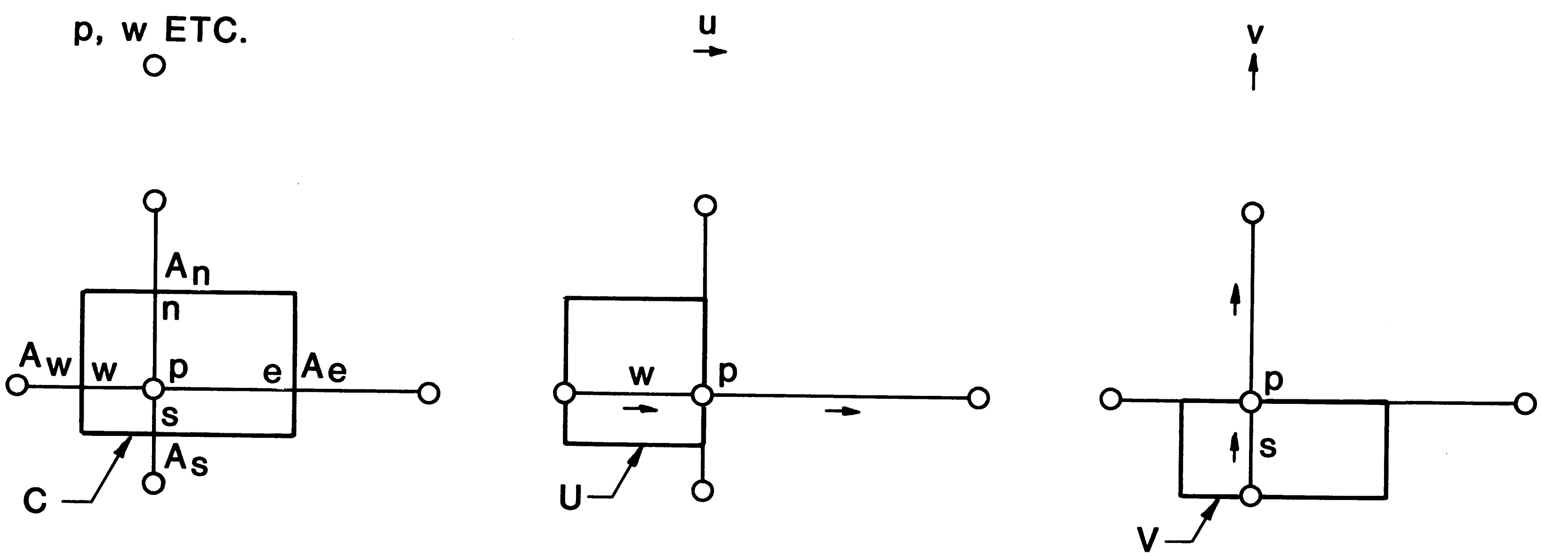
where S_p^ϕ and S_U^ϕ are tabulated in Table 2 and subscripts n, s, e and w refer to north, south, east and west cell faces.

The additional source terms in the equations for swirling flows may induce instability if one is not careful. Only terms which are always negative are permitted in the SP(I,J) expression in order to aid the convergence of the iterative solution procedure by increasing the diagonal dominance of the equation set to be solved at each iteration. Inspection of the arrangement given in Table 2 reveals that the forms given enjoy this desirable stabilizing



THREE GRIDS : FOR p, w ETC. - AT POSITION MARKED (O)
FOR u VELOCITY - AT POSITION MARKED (→)
FOR v VELOCITY - AT POSITION MARKED (↑)

Fig. 2. Staggered grid and notation for the rectangular computational mesh.



CONTROL VOLUMES C, U, V FACE
AREAS A_n, A_s, A_e AND A_w FOR C, SIMILAR FOR U AND V

Fig. 3. The three control volumes associated with points of the three grids.

Table 2. The Form of the Components of the Linearized Source Term*, The Cell Volume Integral $\int_V S_\phi dV = S_P{}^\phi \phi_P + S_U^\phi$ of Eq. (1).

ϕ	Γ_ϕ	$S_p{}^\phi/V$	$S_U{}^\phi/V$
1	0	0	0
u	μ	0	$S^u - \frac{\partial P}{\partial x}$
v	μ	$-2\frac{\mu}{r^2}$	$S^v + \frac{\rho w^2}{r} - \frac{\partial P}{\partial r}$
w	μ	0	$-\frac{\rho v w}{r} - \frac{w}{r^2}\frac{\partial}{\partial r}(r\mu)$
k	μ/σ_k	$-C_\mu C_D \rho^2 k/\mu$	G
ε	μ/σ_ε	$-C_2\rho\varepsilon/k$	$C_1 C_\mu G\rho k/\mu$

In this table certain quantities are defined as follows:

$$S^u = \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r\mu \frac{\partial v}{\partial x}\right)$$

$$S^v = \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial r}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r\mu \frac{\partial v}{\partial r}\right)$$

$$G = \mu\Big[2\Big\{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial r}\right)^2 + \left(\frac{v}{r}\right)^2\Big\} + \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x}\right)^2$$

$$+ \Big\{r \frac{\partial}{\partial r}\left(\frac{w}{r}\right)\Big\}^2 + \left(\frac{\partial w}{\partial x}\right)^2\Big]$$

*In this table, V stands for the cell control volume and $\mu \equiv \mu_{eff}$.

characteristics. A false source stabilizing trick, which has no effect on the final solution, is used in the form of a false source of magnitude:

$$S_{false} = |\dot{m}_{net}| \, (\phi_P^{old} - \phi_P) \tag{9}$$

That is used in the program although not detailed in Table 2.

For the representation of the convective and diffusive terms over the cell control volume surfaces a hybrid scheme is used,[30] which is a combination of the so-called central and upwind differences. Consider the transport across one face of a control volume. Figure 4 shows the western face of area A_w normal to the x-direction, which lies midway between the grid-points W and P distant δx apart. The contribution c_w to the surface integral by the western face area A_w, for example, is given by

$$C = \begin{cases} (\rho u)_w A_w (\phi_W + \phi_P)/2 - (\Gamma_\phi)_w \, A_w (\phi_P - \phi_W)/\delta x \text{ for } |Pe| < 2 \\ (\rho u)_w A_w \, \phi_W \text{ for } Pe \geq 2 \\ (\rho u)_w A_w \, \phi_P \text{ for } Pe \leq -2 \end{cases} \tag{10}$$

where $Pe = (\rho u)_w \, \delta x / (\Gamma_\phi)_w$ is a cell Peclet number calculated at the western face of the cell. Similar expressions are used at the other three faces of the cell. Details regarding the derivation and superiority of this differencing technique may be found in Appendixes A, B and C.

When the various terms are handled in this manner the following general equation is obtained:

$$a_P^\phi \, \phi_P = \sum_j a_j^\phi \, \phi_j + S_U^\phi \tag{11}$$

where $a_P^\phi = \sum_j a_j^\phi - S_P^\phi$

Σ_j = sum over N, S, E and W neighbors

thus linking each ϕ-value at a point P with its four neighboring values.

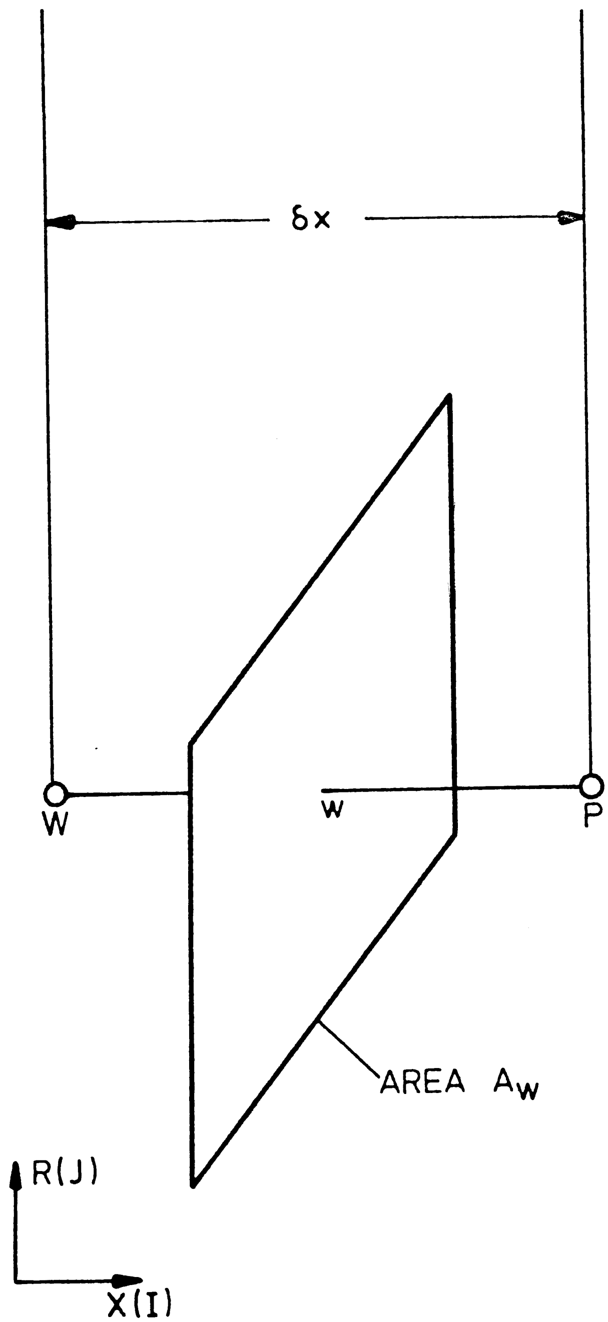


Fig. 4. One face of a control volume across which convective anc diffusive fluxes are considered.

When ϕ = W, k and ε, equations of this type hold for values located at intersections of the grid system, but for ϕ = u and v there are two points of difference: (i) these variables are located to the west and south respectively of the point P and (ii) pressure gradient terms $\partial p/\partial x$ and $\partial p/\partial r$ respectively occur on the right hand side of their equations in addition to the usual source expression. Since the pressure field is not known when the u and v equations are required to be used at any stage of the iteration process, the best estimate so far of the pressure field p* is used. Solving the u and v equations with this p* field gives estimates u* and v* in the flowfield. These p*, u* and v* fields constitute any any iteration stage a first approximation to the solution and they are immediately followed by a succeeding correction at each point P:

$$p_P = p_P^* + p'_P \tag{12}$$

$$u_P = u_P^* + D^u(p'_W - p'_P) \tag{13}$$

$$v_P = v_P^* + D^v(p'_S - p'_P) \tag{14}$$

where $D^u = A_W/a_P^u$

$$D^v = \frac{1}{2}(A_n + A_s)/a_P^v$$

These corrections can be applied only after the pressure correction p' field has been found and this is obtained from a Poisson equation for pressure which may be deduced from a combination of the continuity and momentum equations, resulting in finite difference equations like

$$a_P^p\, p'_P = \sum_j a_j^p\, p'_j + S_U^p \tag{15}$$

where $a_P^p = \sum_j a_j^p$

$\sum_j$ = sum over N, S, E and W neighbors

S_U^p = term which results from mass sources of u* and v* fields.

Thus the application of Eqs. (13) and (14) at any iteration stage brings the first approximation obtained from solution of momentum equations into line with the requirement of continuity - it is here where the continuity equation is used, incorporated in fact in the derivation of the equation for p', Eq. (15).

2.3 Boundary Conditions

There are several methods of numerically positioning an irregular boundary, such as a sloping boundary segment, in finite difference computer programs. In order to retain conceptual simplicity, the stairstep simulation shown in Fig. 5 is utilized. The figure shows an example of grid specification for the geometry under consideration. The flowfield is covered with a nonuniform rectangular grid system. Typically the boundary of the solution domain falls halfway between its immediate nearby parallel gridlines. Clearly, specification of the x and r coordinates of the gridlines, together with information concerned with the position of the sloping sidewall boundary (via specification of JMAX(I) for each I) is sufficient to determine the flowfield of interest.

Up to this point the application of boundary conditions has not yet been considered, and the formulation has been concerned with regular internal mesh points. Insertion of correct boundary conditions requires amendment of the finite difference formulation for the near boundary points.[16] For cells

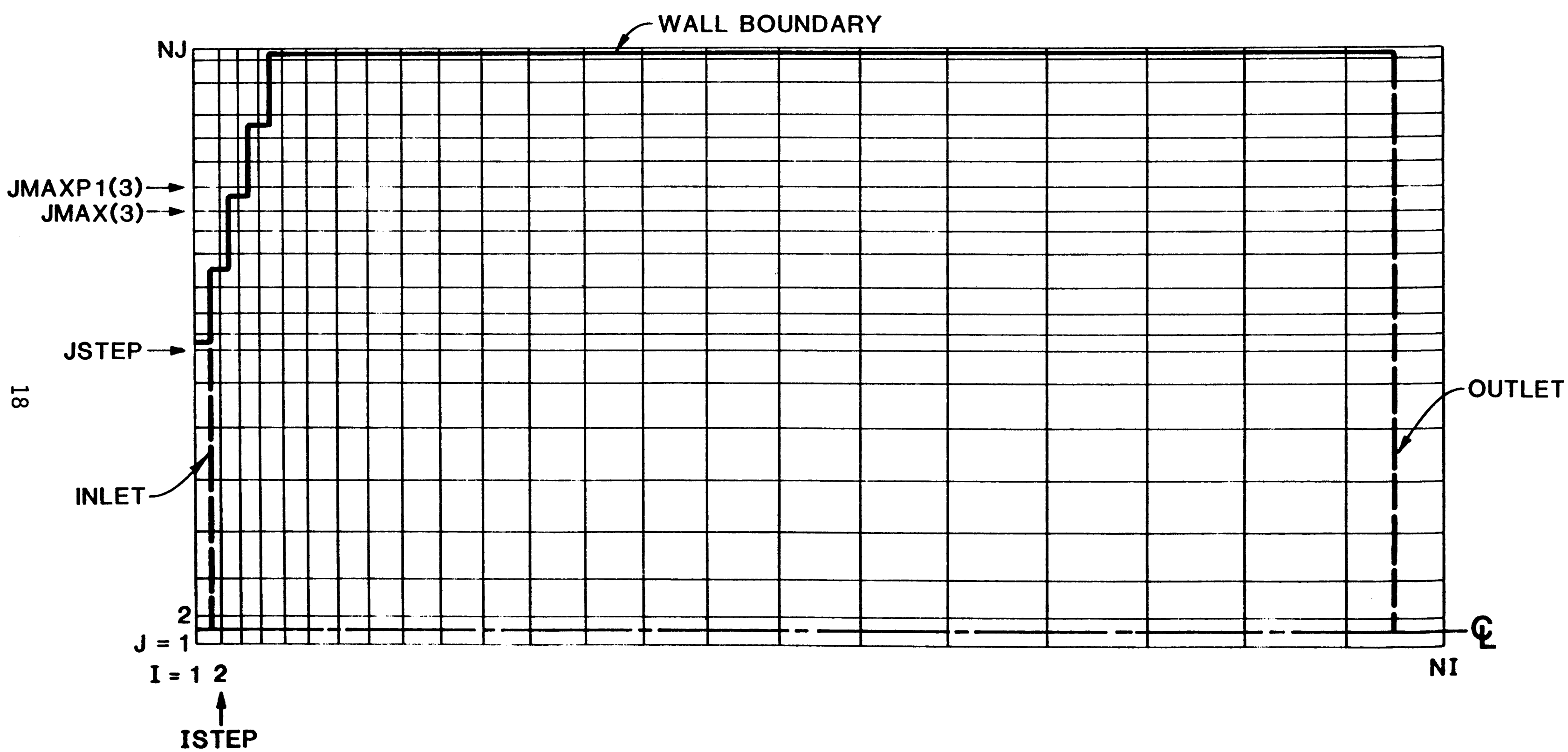


Fig. 5. An example of a grid system being employed to fit the flow domain.

adjacent to the west wall for example, as shown in Fig. 6, there is no convective flux through the cell west boundary which is coincident with an impervious wall. Hence the term for total ϕ-flux through this cell boundary in Eq. (8) is set to zero, and the diffusive flux term $\Gamma_\phi\ \partial\phi/\partial x$ is transferred to the right-hand side of the equation as a false source. In the case of a western wall boundary the normal P-W link is broken by setting:

$$a_W^\phi = 0 \tag{16}$$

and the correct expression is inserted by way of the S_U^ϕ and S_P^ϕ values:

(i) If a value M for the diffusion terms $\Gamma_\phi\ \partial\phi/\partial x$ is specified for the western face of a cell adjacent to a western wall, then as seen in Eq. (8):

$$S_U^\phi = -M/Vol \tag{17}$$

$$S_P^\phi = 0 \tag{18}$$

It is clear that this specification results in correct ϕ balance for the boundary cell, and that if zero normal gradient is the required condition (M = 0) then Eq. (16) is sufficient in itself.

(ii) If the value of ϕ on the boundary ϕ_B is specified, for a western boundary for example, then:

$$S_U^\phi = a_W'\ \phi_B \tag{19}$$

$$S_p^\phi = -a_W' \tag{20}$$

where

$$a_W' = (\Gamma_\phi)_w A_w/\delta x$$

It is clear that this specification gives the correct gradient-diffusion flux out of the western boundary of the cell.

(iii) If ϕ_P is required to be fixed at a value of ϕ_F then:

$$S_U^\phi = \phi_F . 10^{30} \tag{21}$$

$$S_p^\phi = - 10^{30} \tag{22}$$

to that these terms dominate in the equation for ϕ_P with solution $\phi_P = \phi_F$.

(iv) Velocities tangential to a wall boundary usually have boundary conditions of the type v_B given, τ_B(= tangential shear stress) given, or $\tau_B = \alpha v_p + \beta$ (from drag law for example). Any of these may be inserted by the usual treatment (break link and insert via linearized source).

(v) Velocities normal to a wall boundary are given zero values on the wall, but because this leads to incorrect gradient calculations for the first interior point away from the wall, zero normal gradient conditions are applied via breaking the appropriate coupling coefficient in the finite difference equation for this first interior point.

The above discussion about boundary condition types (i) - (v) has been concerned with a western wall boundary shown in Fig. 6. Analogous discussion is appropriate with a northern wall boundary shown in Fig. 7.

Conditions must be specified on the entire boundary around the solution domain, and special novelties concern the pressure correction p', velocity components and the two turbulence quantities k and ε. At the inlets all variables are given definite fixed values (Dirichlet conditions) and at the outlet all variables are given zero normal gradient $\partial\phi/\partial x = 0$ (Neumann conditions) via

$$a_E^\phi = 0 \tag{23}$$

except radial velocity v which is set to zero.

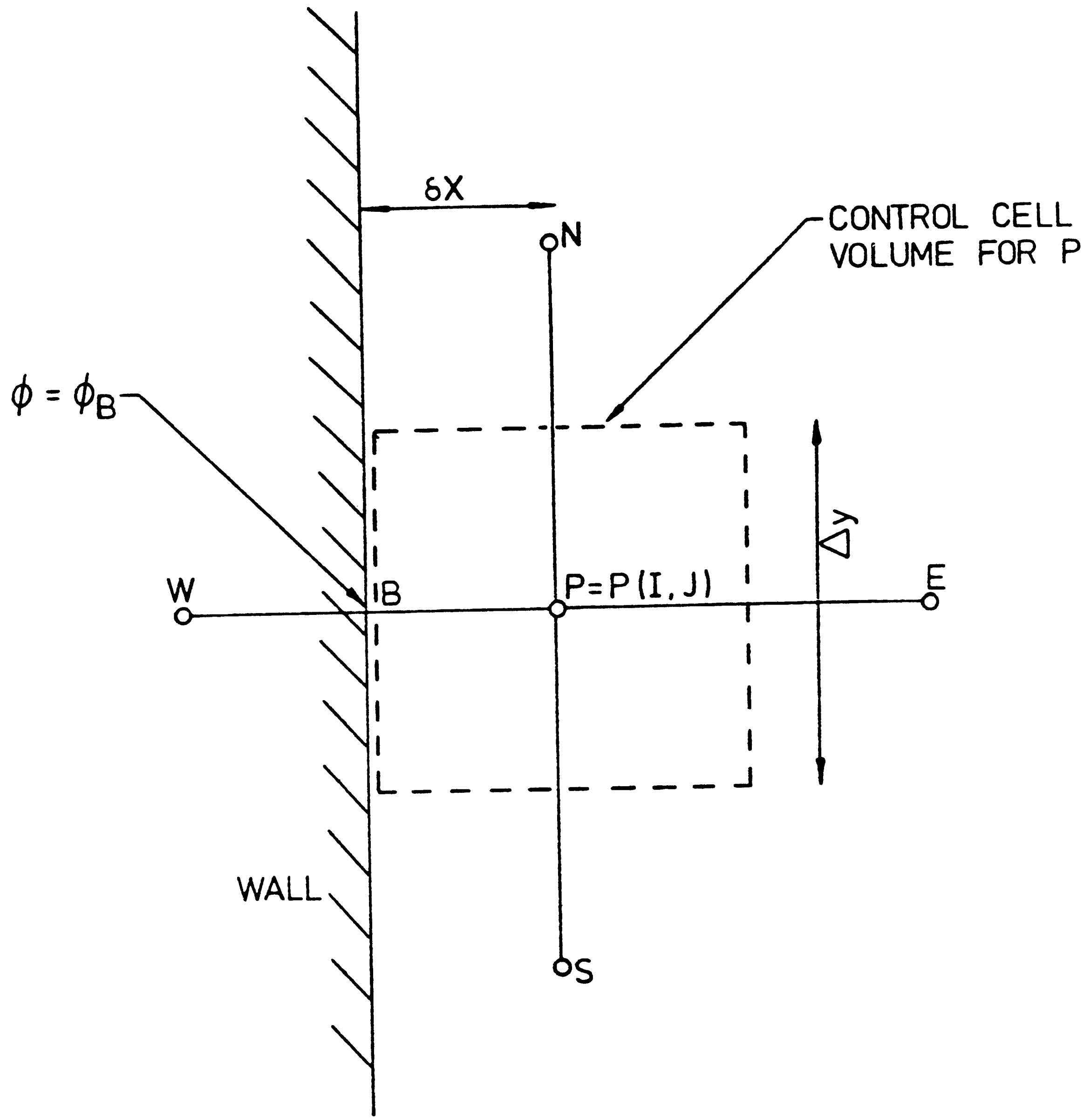


Fig. 6. Wall ϕ-value prescription for western wall.

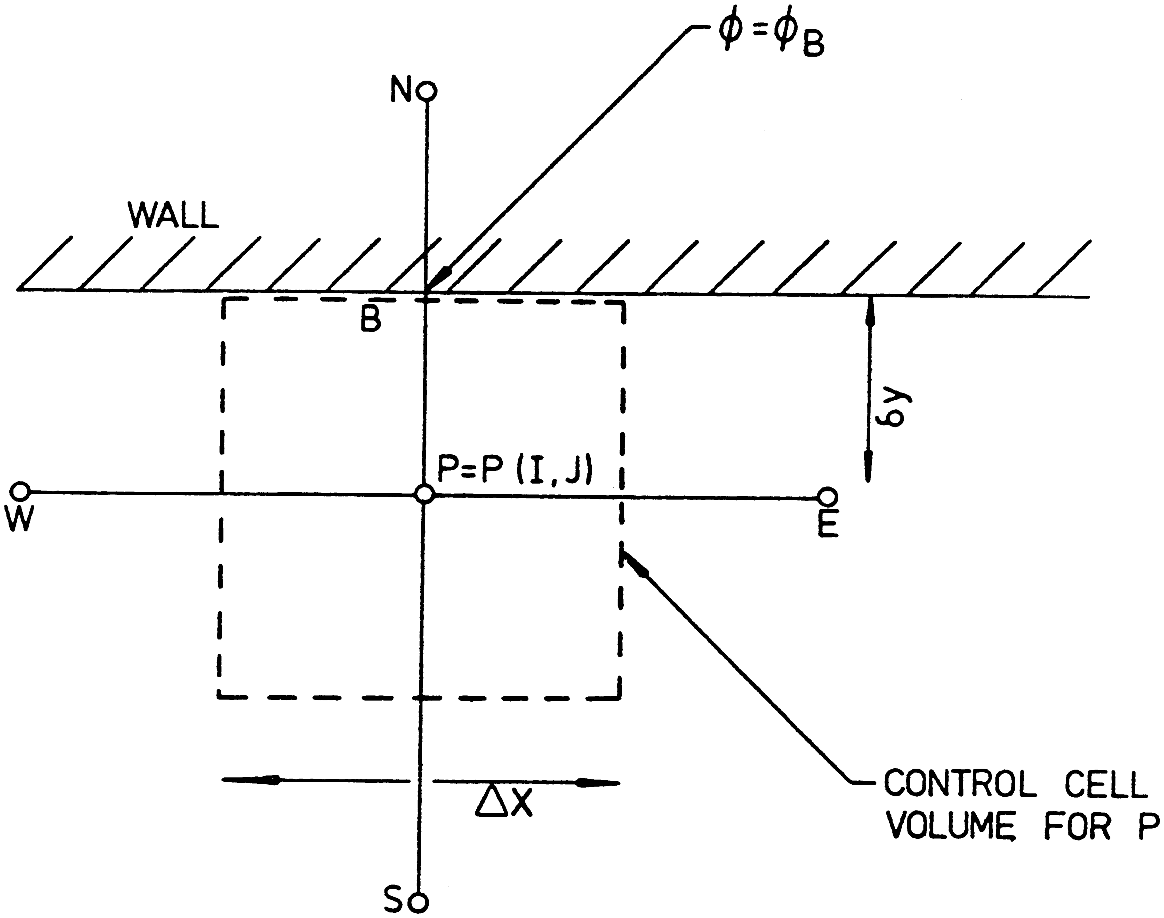


Fig. 7. Wall ϕ-value prescription for northern wall.

The u velocities are then readjusted slightly at each iteration so as to ensure overall mass conservation as compared with the inlet flow. At the symmetry axis $\partial\phi/\partial r = 0$ is assumed for all variables except v (which is given a definite zero value) and w (which is given an implied zero value as follows. Based on measurements[31] of swirl velocity, the BC for w along the centerline specifies solid body rotation through linear interpolation of w(I,2) between the latest value of w(I,3) and zero for w on the centerline. Also p' is given zero normal gradient specification everywhere except at the inlet, where it has zero value: as far as the aerodynamics is concerned it is immaterial what the absolute pressure is, only pressure gradients entering the equations.

2.4 Wall Functions and the Effect of Swirl

To avoid the need for detailed calculations in the nearwall regions, equations are introduced to link velocities, k and ε on the wall to those in the logarithmic region. This one-dimensional Couette flow characterization of the flow (diffusion perpendicular to the wall is dominant) is extremely useful. Also it provides a way around this region of steep nonlinear variation of the variables and the fact that laminar and turbulent effects become of the same order of magnitude. The new equations introduced, called 'wall functions', are introduced and used in the finite difference calculations at near-wall points. They occur in the momenta equations and k-generation terms, and their implementation is discussed elsewhere[11,12] together with appropriate near-wall ε specification.

The effect of swirl on wall function specification is handled as follows: The previous ideas are extended to find total tangential wall shear stress near boundaries [involving x and θ directions for an r = constant wall, and r and θ directions for an x = constant wall]. Then appropriate components are deduced directly [for the u and w velocities which are tangential to an r = constant wall, and v and w velocities which are tangential to an x = constant wall]. The effects on u, v and w momentum equations are incorporated via the usual linearized source technique.

Northern Wall. The applicability of wall function formulations for swirl flows, especially for swirl momentum, appears uncertain a priori. However, from boundary layer velocity measuremetns in turbulent, swirling, pipe flows Backshall and Landis[31] showed that total tangential velocity near the northern top wall $V = (u^2 + w^2)^{1/2}$ is correlated by the universal velocity profile

$$V^+ = (1/\kappa)\ \ell n\ (E\ Y^+) \tag{24}$$

where κ and E are constants. The dimensionless total velocity V^+ and total distance Y^+ are obtained by nondimensionalizing with respect to the total shear velocity $(\tau_t/\rho)^{1/2}$. Hence the total tangential north wall shear stress $\tau_t = (\tau_{rx}{}^2 + \tau_{r\theta}{}^2)^{1/2}$ is obtained from

$$V(\tau_k\rho)^{\frac{1}{2}}/\tau_t = (1/\kappa)\ \ell n[Ey(\tau_k/\rho)^{\frac{1}{2}}/\nu] \tag{25}$$

where τ_k is an approximation for τ_t very near the wall. The quantity τ_k is formulated by observing that convection and diffusion of turbulence kinetic energy are nearly always negligible in this region.[11] Deleting these terms from the k-transport equation and invoking isotropic viscosity μ_{eff} leads to the result

$$\tau_k = (C_D\ C_\mu)^{\frac{1}{2}}\rho k \tag{26}$$

Thus one obtains

$$\tau_t = - V_p \kappa \rho C_\mu^{\frac{1}{4}} C_D^{\frac{1}{4}} k_P^{\frac{1}{2}} / \ln(E\ Y_P^+) \tag{27}$$

where the negative sign is inserted since τ_t and V_p must have opposite directions. The quantities with subscript P are evaluated at the appropriate near-wall points as shown in Fig. 7 for this case of a northern wall.

The north wall diffusion flux for the x-momentum equation is $\mu_{eff}\ \partial u/\partial r$. The τ_{rx} component of τ_t is given by

$$\tau_{rx} = \mu_{eff} \left(\frac{\partial u}{\partial r} + \frac{\partial v}{\partial x}\right) \tag{28}$$

however $\partial v/\partial x$ approaches zero near the north wall.[17] Hence τ_{rx} is the required diffusion flux term to be included as a false source for north wall u-cells; its wall function expression asserts the required boundary influence by indirectly incorporating the boundary condition $u_B = 0$. The above wall function formulation, Eq. (27), is multiplied by the factor $\cos\theta$ ($= u/V$) to obtain τ_{rx}. where θ is the angle between the total tangential velocity vector near the wall and the axial velocity vector ($\theta = \arctan w/u$). This is valid since θ is essentially constant near the wall.[32] Thus this wall function for u is

$$\mu_{eff} \frac{\partial u}{\partial r} = [-\kappa \rho C_\mu^{\frac{1}{4}} C_D^{\frac{1}{4}} k_P^{\frac{1}{2}} / \ln(E\ Y_P^+)] u_P \tag{29}$$

which is precisely the same expression found for nonswirling flows in TEACH, although different values result due to swirl effects on k_p. Also, this result is consistent with an expression obtained by Date.[33]

For w-cells along the north wall, the diffusion is $\mu_{eff}\ \partial w/\partial r$. This quantity appears in the expression for $\tau_{r\theta}$ which is

$$\tau_{r\theta} = \mu_{eff}\ (\frac{\partial w}{\partial r} - \frac{w}{r}) \tag{30}$$

Applying the factor sin $\theta(=w/V)$ to the τ_t wall function yields the following expression for $\tau_{r\theta}$:

$$\tau_{r\theta} = [-\kappa\rho C_\mu^{\frac{1}{4}} C_D^{\frac{1}{4}} k_P^{\frac{1}{2}}/\ell n(E\ Y_P^+)]w_P \tag{31}$$

Hence the north wall function for w is given by

$$\mu_{eff}\ \frac{\partial w}{\partial r} = [-\kappa\rho C_\mu^{\frac{1}{4}} C_D^{\frac{1}{4}} k_P^{\frac{1}{2}}/\ell n(E\ Y_P^+)] + \mu_{eff}/r]w_P \tag{32}$$

<u>Western wall</u>. Momentum wall functions along the west wall are similarly formulated. The total tangential velocity is now $V = (v^2 + w^2)^{1/2}$, and τ_{xr} reduces to $\mu_{eff}\ \partial v/\partial x$. Equation (27) for τ_t is again obtained and the τ_{xr} wall function is formulated by employing the factor cos ϕ; ϕ is the angle between the total tangential velocity vector and the radial velocity vector (ϕ = arctan w/v). Thus one obtains the v-momentum expression:

$$\mu_{eff}\ \frac{\partial v}{\partial x} = [-\kappa\rho C_\mu^{\frac{1}{4}} C_D^{\frac{1}{4}} k_P^{\frac{1}{2}}/\ell n(E\ Y_P^+)]v_P \tag{33}$$

The w-equation wall function on the western wall approximates $\mu_{eff}\ \partial w/\partial x$, which is $\tau_{x\theta}$. Here the factor sin ϕ is used to obtain the necessary wall function as

$$\mu_{eff}\ \frac{\partial w}{\partial x} = [-\kappa\rho C_\mu^{\frac{1}{4}} C_D^{\frac{1}{4}} k_P^{\frac{1}{2}}/\ell n(E\ Y_P^+)]w_P \tag{34}$$

Other details. In accordance with Gosman and Pun,[17] if Y^+ for a cell adjacent to any solid boundary is less than 11.63, it lies within the laminar sublayer where the laminar viscosity dominates. Hence, the wall function employed for north wall u-cells, for example, is

$$\mu_{eff} \frac{\partial u}{\partial r} = -\mu_\ell u_P/\delta_P \tag{35}$$

where $u_B = 0$ has been employed. Similar expressions are analagously obtained for other momentum diffusion terms of the problem.

For turbulence energy, from Eq. (30)

$$k\rho/|\tau_B| = (C_\mu C_D)^{-\frac{1}{2}} = \text{constant} \tag{36}$$

near a solid boundary which implies that the wall flux is zero. Accordingly, zero normal gradient prescription for k is appropriate for wall cells via Eq. (16). Also the shear stress wall functions are employed for the k-generation source G for k-cells along a wall. The general expression for G, given in Table 2, can alternately be expressed as

$$G = 2\mu_{eff}[(\frac{\partial u}{\partial x})^2 + (\frac{\partial v}{\partial r})^2 + (\frac{v}{r})^2] + [\tau_{rx}^2 + \tau_{r\theta}^2 + \tau_{x\theta}^2]/\mu_{eff} \tag{37}$$

For north wall cells $\tau_t^2 = \tau_{rx}^2 + \tau_{r\theta}^2$ which yields

$$G = 2\mu_{eff}[(\frac{\partial u}{\partial x})^2 + (\frac{\partial v}{\partial r})^2 + (\frac{v}{r})^2] + \tau_t^2/\mu_{eff} + \mu_{eff}(\frac{\partial w}{\partial x})^2 \tag{38}$$

where the previous wall function expression for τ_t is employed. Similarly, the result for west wall cells is

$$G = 2\mu_{eff}[(\frac{\partial u}{\partial x})^2 + (\frac{\partial v}{\partial r})^2 + (\frac{v}{r})^2] + \tau_t^2/\mu_{eff} + \mu_{eff}(\frac{\partial w}{\partial r} - \frac{w}{r})^2 \quad (39)$$

where $\tau_t^2 = \tau_{xr}^2 + \tau_{x\theta}^2$.

As for energy dissipation, near a wall the length scale is assumed to be proportional to the normal distance from the wall which leads to

$$\varepsilon = \frac{(C_\mu C_D)^{3/4}}{C_D \kappa} \frac{k_P^{3/2}}{\delta_P} \quad (40)$$

This is the effective wall boundary condition on ε and in the code this value is fixed for near wall points via the linearized source procedure, using Eqs. (21) and (22).

2.5 Boundary Conditions Summary

Table 3 summarizes the boundary values used in the present study, which are incorporated into the computer program. The symbol N in the table signifies that zero normal gradient prescription is appropriate. Footnotes (a) to (g) are dealt with as follows:

(a) Inlet values UIN etc. are directly specified and easily altered.

(b) The axial velocities at the outlet are deduced from their immediate upstream values by adding a fixed amount of each of U(NIM1,J) (J = 1 to NJM1); this amount is calculated to ensure overall mass conservation.

(c) Near wall tangential velocities are connected with their zero wall values by way of the tangential shear stress wall functions.

(d) Near-axis swirl velocity at J = 2 is fixed for solid body rotation in terms of the previous w at J = 3 and its zero value on the axis.

(e) Outlet values of k and ε are unimportant because of the limited upstream influence at high Reynolds numbers and zero gradient is

Table 3. Boundary Conditions (N stands for zero normal gradient specified)

(Footnotes are detailed in section 2.3 of the text)

Algebraic Variable	Inlets[a]	Outlet	Top Wall	Side wall	Symmetry Axis
u	UIN	adjusted[b]	(c)	N	N
v	0	0	N	(c)	0
w	WIN	N	(c)	(c)	0[d]
κ	TEIN	N[e]	N	N	N
ε	EDIN	N[e]	(f)	(f)	N
p'	0[g]	N	N	N	N

specified. If the local Peclet number is greater than +2, there is no upstream influence because of the implications of the hybrid formulation of convection and diffusion, and the downstream convection dominates.

(f) Near wall ε values are <u>fixed</u> using the length scale near the wall and the current value of k.

(g) The pressure correction p' possesses <u>zero normal gradient</u> specification everywhere except the inlet. Just one internal pressure p specification at (2,2) is then required to allow all the pressures to be calculated.

This concludes the discussion of boundary condition specification; its implementation is returned to in Section 4 of the present text.

2.6 Solution Procedure

The finite difference equations and boundary conditions constitute a system of strongly-coupled simultaneous algebraic equations. Though they appear linear they are not since the coefficients and source terms are themselves functions of some of the variables, and the velocity equations are strongly linked through the pressure. The solution proceeds by the cyclic repetition of the following steps:

(i) Guess the values of all variables including p^*. Hence calculate auxiliary variables like turbulent viscosity etc.

(ii) Solve the axial and radial momentum equations to obtain u^* and v^* from equations like Eq. (11).

(iii) Solve the pressure correction equation, Eq. (15), to obtain p'.

(iv) Calculate the pressure p and the corrected velocities u and v from Eqs. (12) through (14).

(v) Solve the equations like Eq. (11) for the other ϕ's successively.

(vi) Treat the new values of the variables as improved guesses and return to step (i); repeat the process until convergence.

In the solution procedure algebraic equations like Eq. (11) are solved many times, coefficient and source updating being carried out prior to each occasion. The practice used here is to make use of the well-known tridiagonal matrix algorithm (TDMA), whereby a set of equations, each with exactly three unknowns in a particular order except the first and last which have exactly two unknowns, may be solved sequentially. One considers the values at grid-points along a vertical gridline to be unknown (values at P, N and S for each point P), but take as known, most recent values being used, the values at each E and W neighbor. The TDMA is then applied to this vertical gridline. In this manner one can traverse along all the lines in the vertical direction sequentially from left to right of the integration domain.

Two points clarify the solution technique regarding the stairstep sloping boundary. Firstly, interior points adjacent to the boundary must 'feel' the boundary in the usual way. Thus values in the external field must not be inadvertently picked up at these points. Usually the standard coupling coefficient of interest is set to zero and the wall effect given by way of a false source, according to the previous description. Secondly, if the TDMA is applied in columns over the entire 2-D array of points, the solution at external points can be automatically forced to zero by giving S_p^ϕ a very large negative value at these locations. This term then dominates in its finite difference equation at P with solution $\phi = 0$. Alternatively, the domain of TDMA execution can be restricted to the physically meaningful portion of the 2-D array of points. The latter technique is incorporated into the present computer code.

At each iteration it is necessary to employ some degree of under-relaxation when solving equations like Eq. (11). A weighted average of the newly calculated value and the previous value is taken at each point. If $\tilde{\phi}_p^{\;n+1}$ represents the newly calculated value of ϕ at the n+1 iteration, it is obtained from

$$a_P^\phi \, \tilde{\phi}_P^{\;n+1} = \sum_j a_j^{\;\phi} \phi_j^{\;n} + S_U^\phi \tag{41}$$

The notation and similarity to Eq. (11) is obvious and

$$\phi_P^{\;n+1} = f \, \tilde{\phi}_P^{\;n+1} + (1-f) \, \phi_P^{\;n} \tag{42}$$

where $\phi_p^{\;n+1}$ represents the underrelaxed value of ϕ_p, employing the no-relaxation value $\tilde{\phi}_p^{\;n+1}$ and the underrelaxation parameter f ($0 < f \leq 1$). Rather than calculate and store the $\tilde{\phi}^{n+1}$ values and then apply Eq. (42), it is more convenient to modify and use Eq. (41) directly in the form

$$(a_P^\phi/f)\phi_P^{\;n+1} = \sum_j a_j^\phi \phi_j^{\;n} + [S_U^\phi + (1-f)\, \frac{a_P^\phi}{f}\, \phi_P^{\;n}] \tag{43}$$

so that the underrelaxed value is calculated immediately. This is the version used in the present code.

The effect of underrelaxation factor values on convergence rate is considerable. Unacceptably slow convergence or divergence of the solution is obtained if the factors are too low or too high, respectively. Large pressure corrections arise which produce large u- and v-velocity corrections. If these corrections are too large per iteration, the nonlinearity of the FDEs causes divergence.

Velocity and pressure corrections per iteration become smaller as the solution proceeds toward convergence. Thus the underrelaxation factors (especially for u-and v-velocities) have small initial values, to prevent divergence, and increase for faster convergence as the corrections become smaller. Hence the underrelaxation factors for u, v and w are increased linearly during the first 40 iterations.

Increased accuracy of the initial estimate of field variables clearly reduces the amount of computational work required. Also it yields a slower rate-of-variable change and smaller corrections for continuity, which promotes convergence.

Since the user generally desires to utilize such a computer code to solve a series of problems, as in a parametric study for example, he almost always has converged (or partially-converged) solutions from previous computer runs of a similar problem. By storing field values of all variables on disk upon convergence for each swirl loop, a much better initial estimate conveniently becomes available for corresponding swirl cases of future problems, or for more stringent convergence of the same problem.

Final convergence is decided by way of a residual-source criterion, which measures the departure from exactness for the variable ϕ at the point P. The residual sources are defined for each variable at each point by equations like

$$R_P^\phi = a_P^\phi \phi_P - \sum_j a_j^\phi \phi_j - S_U^\phi \tag{44}$$

When each of these quantities becomes smaller than a certain fraction of a reference value, the finite difference equations are declared to be solved. Typically, the solution is considered to be converged if the cumulative sum of the absolute residuals throughout the field for <u>all</u> variables is less than 0.5 per cent of the inlet flow rate of the corresponding variable.

3. THE STARPIC COMPUTER PROGRAM

3.1 Outstanding Features

The case described. The computer program code described and listed performs the function outlined in Section 2 for the particular 2-D axisymmetric geometry shown schematically in Fig. 5. The flowfield is covered with a gradually expanding rectangular grid system. The quantity EPSX specifies the degree of grid expansion in the axial direction, and in the radial direction it is determined through specification of the radial position of each individual horizontal grid line. The boundary of the solution domain falls halfway between its immediate nearby parallel gridlines; and clearly specification of JMAX(I) for each I (see Fig. 5) is sufficient to determine the flowfield of interest. This provides a stairstep approximation to any monotonically increasing top wall radius as required so as to simulate any sudden expansion, sloping straight or curved sidewall. The code is equipped to handle any of these with ease. The program is an advanced version of a framework of ideas embodied into the 1974 Imperial College TEACH computer program,[17] from which the present work has been developed. The following discussion is a commentary on the final developed program. A listing is supplied as Appendix E and should be read simultaneously with the discussion.

General arrangement. The STARPIC program (see the flow chart of Fig. 8) has been written (in Fortran 4) primarily with a view to maximizing simplicity and ease of amendment at the expense of computer time. This is achieved by limiting to the MAIN subprogram those features which characterize the particular flow conditions being investigated, while different boundary conditions may be inserted by way of the PROMOD (problem modifications) subroutine. The various functions of these and other subroutines are given briefly in Table 4 and in detail in sections 3.2 to 3.12. MAIN is the section of the

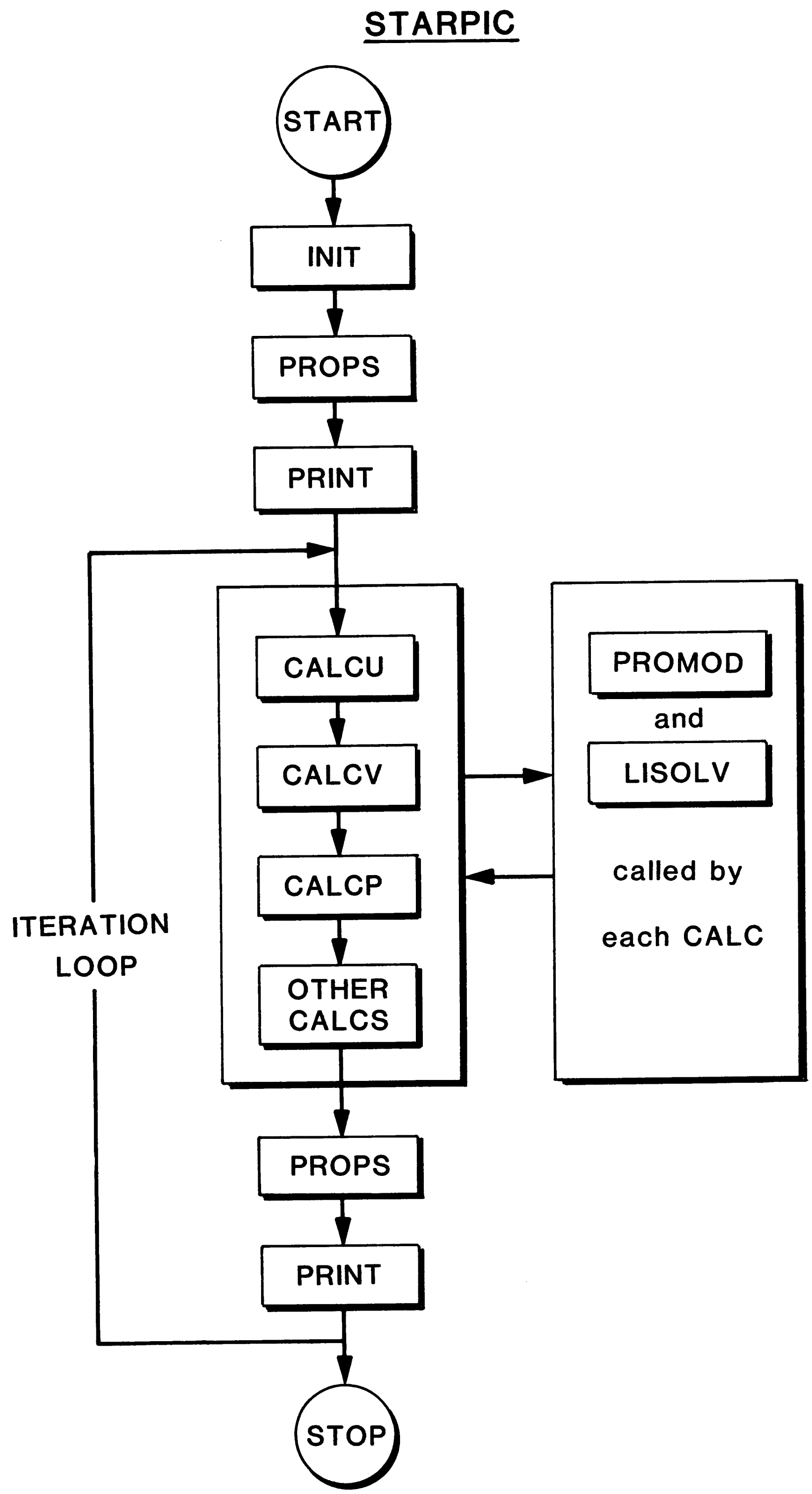


Fig. 8. Flow chart of computer program showing the subroutine arrangement.

Table 4. Subroutine Tasks

Subroutine	Task
MAIN	Controls and monitors the entire sequence of calculations: initialization, properties and initial output; the iteration loop with calls to update main variables, other mixture properties and intermediate output; and, after termination of the iteration loop, final output, an increment in inlet degree of swirl and a return to the beginning again.
INIT	Sets values to the numerous geometric quantities concerned with grid structure, see Figs. 9-11, and initializes most variables to zero or other reference value.
PROPS	Updates the fluid properties via calculation of turbulent viscosity, under-relaxed using its previous value. In nonisothermal flows, perhaps with chemical reaction, additional species' mass fractions, temperature and density are also calculated here, with an appeal to PROMOD (1) for any other modifications.
PRINT	Prints out an entire variable field according to a standard format.
CLACU and CALCV	Calculates coupling coefficients of finite difference equation for axial velocity u* and radial velocity v*, calls PROMOD (2) and PROMOD (3) for boundary modifications and LISOLV for entire field of variables to be updated to get u* and v* fields.

Table 4 (continued)

CALCP	Calculates coupling efficients of finite difference equation for pressure correction p'; calls PROMOD (4) for boundary modifications and LISOLV to obtain p' field. The subroutine closes with p*, u* and v* being 'corrected' with p', u' and v'.
Other CALC Subroutines	Calculates coupling coefficients of appropriate finite difference equation, calls appropriate part of PROMOD and then LISOLV for complete update of the variable in question.
PROMOD	Modifies the values of the finite difference equation coefficients, or the variables, near walls or other boundaries where particular conditions apply. The subroutine is divided into chapters, each handling a particular variable and being called from a CALC subroutine, and each chapter considers all the boundaries around the solution domain.
LISOLV	Updates entire field of a particular variable, by applying TDMA (tridiagonal matrix algorithm) to all the lines in the r-direction sequentially from left to right of the integration domain.

program to which a user will devote most of his attention, provided that he wishes to investigate configurations easily derivable from that shown in Fig. 5. He may also be concerned with PROMOD to adjust boundary condition types. Moreover all the subprograms are divided into chapters, each with a prescribed task to accomplish, thus decreasing the possibility of error on problem specification and modification.

MAIN controls the entire iterative solution procedure, with initial calls to INIT (initialization), PROPS (properties) and PRINT (for output of all variable fields). Repeated intermediate calls occur during the iteration loop to the CALC subroutines (CALCU for updating calculation of the U-velocity field, etc.), PROPS and PRINT (when required). Finally, after convergence or MAXIT (maximum iterations) number of iterations, PRINT is called again, together with the calculation and output of any special quantities required which are deducible from the flowfield values. Each of the CALC subroutines calls PROMOD (problem modifications) for modifications to the usual formulation because of boundary conditions. Subroutine LISOLV (line solve) is also called by each CALC subroutine to make several update sweeps of the relevant field of variables by applying the TDMA (tridiagonal matrix algorithm) to all the lines in the r-direction sequentially from left to right of the integration domain (NSWPU times for the u-field, etc. - number of sweeps for u).

Major variables. The discussion of Fortran variables is restricted to those symbols enjoying major significance in the accompanying listing of the program in Appendix E. Other symbols will readily yield their meanings on inspection of the context and their memonics; or they appear in parts of the program which the user is enjoined not to disarrange, the user restricting most of his attention to MAIN and PROMOD. A glossary of Fortran symbols is given in Appendix D. Table 5 lists principal dependent variables and controlling parameters.

Table 5. Principal Dependent Variables and Controlling Parameters

Algebraic Variable	Fortran Variable	Alphanumeric Heading	Logical Variable	Inlet Value	CALC Subroutine	Underrelaxation factor	Number of sweeps of LISOLV per iteration	Prandtl/ Schmidt Number	Residual Source Term
u	U	HEDU	INCALU	UIN	CALCU	URFU	NSWPU	-	RESORU
v	V	HEDV	INCALV	VIN	CALCV	URFV	NSWPV	-	RESORV
w	W	HEDW	INCALW	WIN	CALCW	URFW	NSWPW	PRW	RESORW
k	TE	HEDK	INCALK	TEIN	CALCTE	URFK	NSWPK	PRTE	RESORK
ε	ED	HEDD	INCALD	EDIN	CALCED	URFE	NSWPD	PRED	RESORE
p'	PP	HEDP	INCALP	-	CALCP	URFP	NSWPP	-	(RESORM for mass calculated in CALCP)
μ	VIS	HEDVIS	INPRO	-	PROPS	URFVIS	-	-	-

3.2 The MAIN Subprogram

General arrangement. MAIN is that section of the program to which a user will devote most of his attention. The function of MAIN has been described in section 3.1. It is divided into chapters, each with a specific task, and a description of the individual chapters of MAIN now follows.

CO Preliminaries. Dimension, common and data blocks are followed by user input logical variables, which activate (when specified as true) certain special features of the program. As discussed in detail in Chapter 4, IFINE specifies a fine mesh in the x-direction with NI=35 as opposed to a coarser mesh with NI=23; IWRITE writes all dependent variables onto allocated disk storage; NONDIM calculates and prints the nondimensional solution in addition to the dimensional one; IREAD reads from allocated disk storage the initial solution guess, which is a previous solution of a similar problem; INITAL prints the initial guess of the solution; and INPLOT produces a line-printer plot of streamlines. Specification of NSWPU (number of sweeps for U), etc. and input read statements for alphanumeric headings HEDU (heading for U velocity), etc. are also located here. These alphanumeric headings are the only input data which appear after the program and are listed here in order:

U VELOCITY

V VELOCITY

W VELOCITY

PRESSURE

TEMPERATURE

TURBULENCE ENERGY

TURBULENCE DISSIPATION

VISCOSITY

DIMENSIONLESS LENGTH SCALE

DIMENSIONLESS STREAM FUNCTION

RADIAL COORDINATE OF STREAMLINES

DIMENSIONLESS U VELOCITY

DIMENSIONLESS V VELOCITY

DIMENSIONLESS W VELOCITY

DIMENSIONLESS PRESSURE

DIMENSIONLESS TURBULENCE ENERGY

DIMENSIONLESS STREAMLINE COORDS

DIMENSIONLESS EFF. VISCOSITY

C1 Parameters and control indices. This chapter really defines the problem to be solved. It is implicity that the problem is axisymmetric in cylindrical coordinates with the setting of INDCOS = 2. The problem domain is specified by dealing with the grid exemplified in Fig. 5, and giving values to the key parameters there-in. Other matters include dependent variable selection (setting .TRUE. or .FALSE. to INCALU etc.), fluid properties (see Prandtl/Schmidt numbers of Table 4), turbulence constants, boundary values (see inlet values of Table 4), pressure calculation reference point (IPREF, JPREF) and program control and monitor.

C2 Initial operations. Based on the problem specification already accomplished in C0 and C1, certain other initial operations are needed prior to the iteration procedure. In C2 the geometric quantities for the grids (see Figs. 9-11) are calculated and main 2-D array variables are set to zero or obvious initial values by way of the call to subroutine INIT. Returning to MAIN, the dependent variable fields are first specified at the inlet boundary including swirl velocity which uses VANB or SWNB for flat or solid-body rotation profile according to whether NSBR (number for solid body rotation) equals 0 or 1. Then estimates of dependent variables in the field are calculated, as many of them depend on inlet values. Following the call to PROPS for fluid properties, an initial output is made of certain flow and geometric quantities. Finally, if IREAD = .TRUE. each dependent variable array is read sequentially from the disk storage allocated to unit 12 to re-specify the initial field estimates with improved values. Thus the user should specify IREAD = .TRUE. when a previous solution to a similar problem is available on disk, as this reduces the number of iterations required for convergence.

C3 Iteration loop. This is the section of MAIN that is repeated at each iteration where NITER counts the number of iterations. The task of C3 is several-fold: to adjust the underrelaxation factors for enhanced convergence based on numerical experience; to update dependent variables (via calls to subroutine CALCU if INCALU is true, etc.); to update fluid properties (if INPRO is true the call to PROPS updates secondary dependent variables); to give intermediate output of residual source sums and field values at monitored location (IMON, JMON) for all values of NITER; to include appropriate field prints when NITER=JPRINT; and finally to make appropriate iteration termination tests.

C4 Final operations and output. Having decided to terminate the iteration process, control transfers to C4 to obtain final variable field printouts and to calculate and print nondimensional variables if NONDIM = .TRUE. Also, a call to subroutine STRMFN calculates dimensionless stream function and determines the coordinates of points along NSTLN streamlines for plotting. Further, if IWRITE = .TRUE. each dependent variable array is stored sequentially on the disk space allocated to unit 11.

Control has now finished with the current problem. However, if LFS (loop for swirl) is less than LFSMAX (loop for swirl maximum) then LFS is incremented by 1, and the inlet swirl velocity profile is recalculated for the next swirl problem based on SWNB (swirl number block) or VANB (vane angle block) according to whether NSBR (number for solid body rotation) equals 0 or 1. Before control begins C3 for the new swirl problem, the precious solution of a similar problem will be read from disk storage allocated to unit 12 if IREAD = .TRUE. This improved initial field estimate reduces the number of iterations required for convergence.

3.3 Subroutine INIT

INIT consists of two main chapters:

C1 Calculate geometric coefficients. Geometric quantities for the three grids for C, U and V cell control volumes (see Figs. 3 and 9-11) are here calculated once and for all for the particular geometric system previously specified in MAIN. Notice that setting INDCOS = 1 suppresses the variation of R with Y and thus obtains cartesian coordinates. In the current program, however, the problem is correctly coded only for cylindrical coordinates.

C2 Set variables to zero. In C2 most dependent and other 2-D array variables are set to zero throughout the flowfield although some are set to obvious nonzero initial values.

3.4 Subroutine PROPS

There is only one chapter:

C1 Viscosity. The turbulent viscosity μ is calculated from the two-equation k-ε turbulence model, Eqs. (5) and (6), making use of the URFVIS underrelaxation parameter.

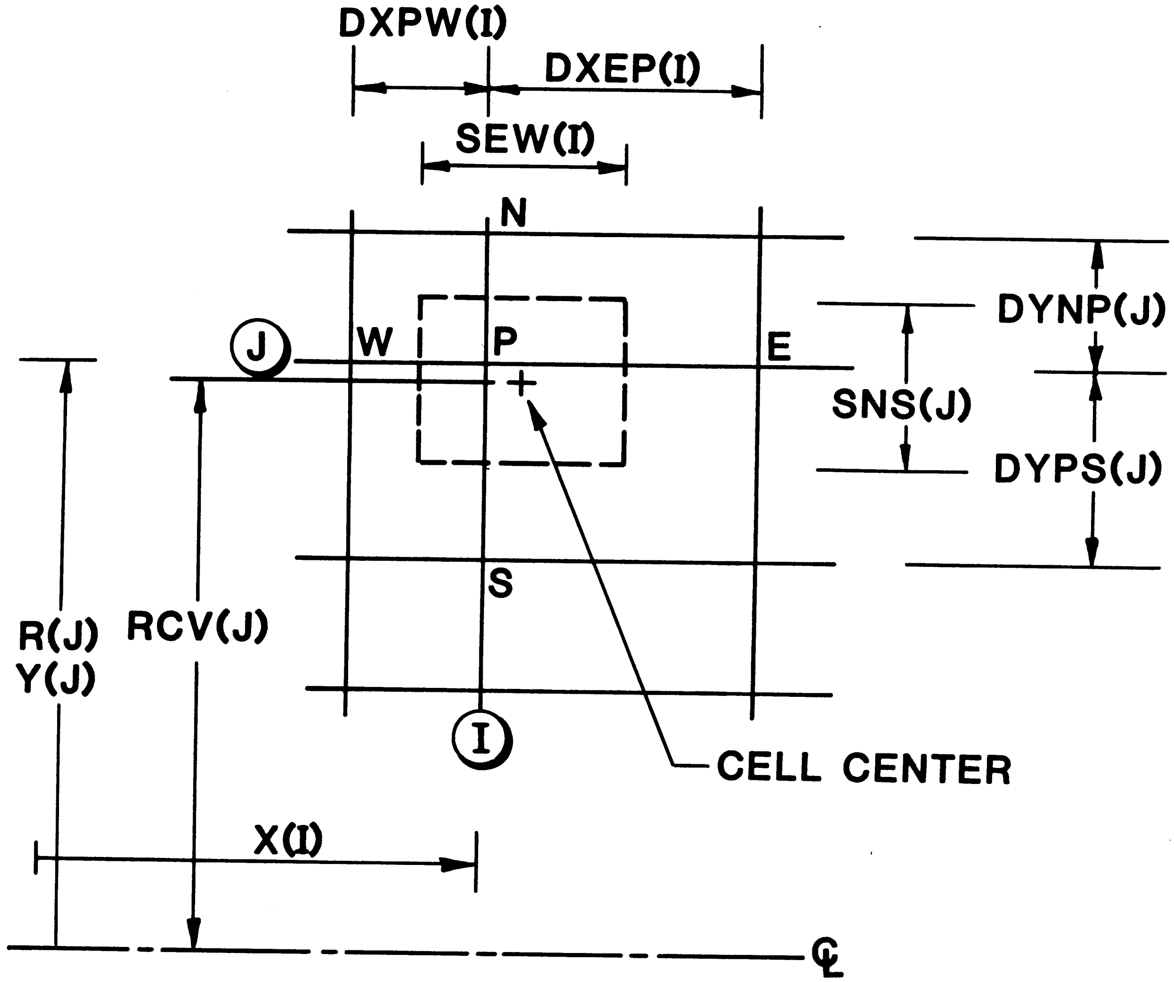


Fig. 9. Fortran variables related to the grid for C cells.

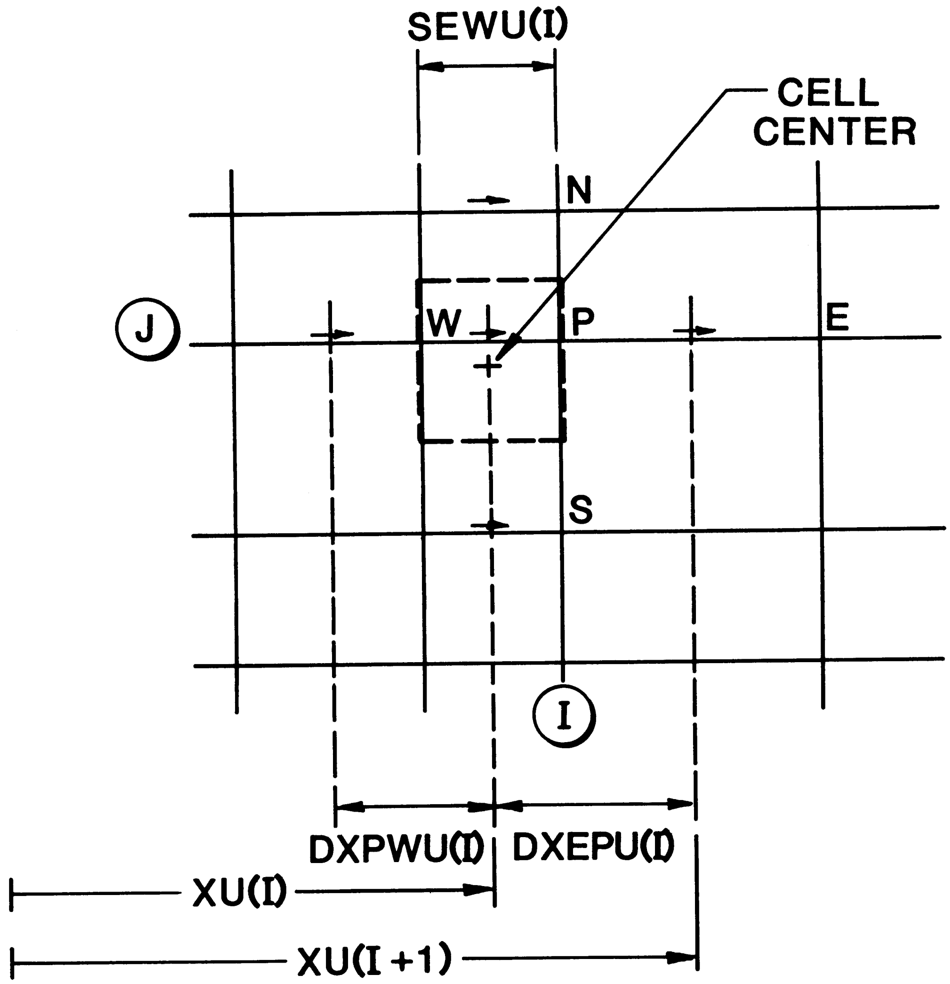


Fig. 10. Fortran variables related to the grid for U cells.

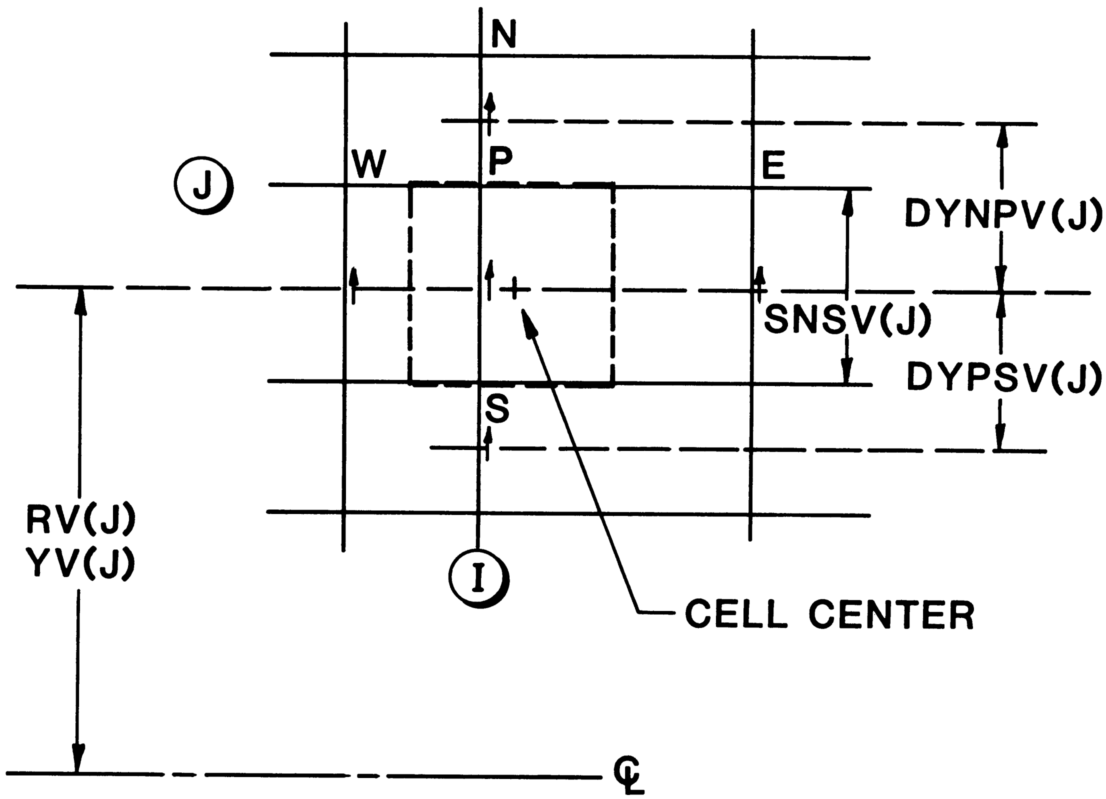


Fig. 11. Fortran variables related to the grid for V cells.

3.5 Subroutines CALCU and CALCV

CALCU has the task of calculating the appropriate coupling coefficients linking the value of U* at each 'w' point of the mesh system with its four neighbors via Eq. (11). The formulation is completely general and correct only for internal points but the call to PROMOD (2)(problem modifications, chapter 2) corrects for near boundary points by applying the correct formulation via a linearized source technique (see section 2.3). Underrelaxation via Eq. (43) precedes the updating of the U* field via a call to LISOLV, this being called NSWPU (number of sweeps for U) times for multiple inner updates of the U* field. The task of CALCV is similar as it renews the V* field and utilizes PROMOD (3). The chapters are:

C1 Assembly of coefficients. Two nested loops for I and J ensure that all internal points of the mesh are considered in C1. For each (I, J) - point the following sequence of calculations is performed:

(i) the cell face areas and volume (utilizing the geometric variables of Figs. 9-11.

(ii) the convection coefficients via the mass-velocities GN etc. and mass flow fluxes through the faces CN etc.

(iii) the diffusion coefficients utilizing viscosities, areas and distances to obtain DN etc.

(iv) the coefficients of source terms - here only the 'false' source stabilizing trick components of Eq. (9) are used; other parts of the source term are added in later.

(v) the main coefficients AN etc. are assembled, using the 'hybrid' formulation of central and upwind differences, Eq. (10). The SU and SP components of the source expression are now determined with the pressure term going into SU, see Tables 1 and 2.

C2 Problem modifications. The general forumlation of C1 is modified for near boundary points via the call to PROMOD. The call is to Chapter 2 of PROMOD in the case of CALCU and to Chapter 3 in the case of CALCV.

C3 Final coefficient assembly and residual source calculation. Two nested loops for I and J allow all points of the grid to be dealt with in turn. Firstly, AP (= a_p) is calculated and RESORU (residual source for U) is incremented, and secondly, the underrelaxation demands of Eq. (43) are applied.

C4 Solution of difference equation. LISOLV is called NSWPU times in order to update the U* field several times (NSWPV times in the case of V*). The LISOLV parameter list begins 3, 2, ... in the case of CALCU, but 2, 3, ... in the case of CALCV. Careful observation of the relationship between the physical and reference storage locations of the displaced grid systems for U and V confirm this requirement.

3.6 Subroutine CALCP

CALCP deals with the variable PP (= p' the pressure correction) in a similar fashion to the CALCU and CALCV subroutines just described. However, there are several points of difference:

(i) the subroutine is concerned with Eq. (15) and the calculation of p',

(ii) the assembly of coefficients is simpler than the previous two subroutines,

(iii) the subroutine closes with the application of the p' field to correct the latest estimates so far of the u, v and p fields, that is u*, v* and p*, by appropriate use of Eqs. (12) through (14).

C1 Assembly of coefficients. Two nested loops sweep through the entire grid system. At each point the following sequence of calculations is performed: areas and volume of cells, coupling coefficients, source terms and absolute mass sources.

C2 Problem modification. A call statement to PROMOD (4) is included.

C3 Final coefficient assembly. Coefficient AP is calculated for all points of the mesh.

C4 Solution of difference equations. LISOLV is called NSWPP times in order to obtain a good estimate of the p' field. The parameter list, beginning 2, 2, ..., is appropriate since pressures are stored at grid intersections.

C5 Correct velocities and pressure. When this stage is reached, first estimates of the u, v and p fields (U* from CALCU, V* from CALCV and p* from previous iteration) are known. Equations (12) through (14) are applied to obtain better estimates and ensure that the U* and V* momentum equation first-estimate solutions are brought into conformity with the continuity equation. Notice that the DO loop ranges are a little complicated so as to operate only on internal field points. There is no underrelaxation here on the velocity components (this was done prior to obtaining the U* and V* fields). Also, the required p' correction is added to the latest estimate of the pressure field p*.

3.7 The other CALC subroutines

The other main dependent variables which are to be solved from their governing partial differential equations are w, k, and ε as seen in Table 1. Each of these is dealt with in a separate subroutine denoted by the name CALC followed by W, TE and ED. They are each called sequentially in this order directly from the MAIN subprogram; and each of them is structured similarly to the CALCU and CALCV subroutines described in section 3.5. Only highlights of the differences need be described.

C1 Assembly of coefficients. Since all these variables are physically located at the grid intersections, both the I and J loops begin with the value 2 and ensure a sweep over all internal mesh points. The third part of C1 is concerned with diffusion coefficients and it is here that the Prandtl/Schmidt numbers PRTE, etc. in Table 4 are brought into play: viscosities are divided by PRTE, etc. in order to obtain the appropriate exchange coefficients. The fourth part concerns source terms and rather longer expressions are formulated as required from Table 1.

C2, C3 and C4. These bear such a strong resemblance to their corresponding chapters already described in section 3.5 that no further elaboration need be given here, except to remark that the LISOLV call always begins with the parameters 2, 2.

3.8 Subroutine LISOLV

LISOLV (line solver) is called from each of the CALC subroutines. When called it provides one complete sweep of the relevant interior points at which the variable is located. Also it replaces the 2-D array PHI, the last parameter of the call list, with the solution of the equations previously built-up in the calling subroutine. This is effected by means of the well-known TDMA (tridiagonal matrix algorithm) being applied to each vertical gridline in turn, and traversing along all such lines in the vertical direction sequentially from left to right of the integration domain.

The subroutine begins with a DO loop for the W-E sweep. There is some complexity here involving ISTART and JSTART, the first two parameters of the call list (these are usually given the values 2, 2 but are 3, 2 and 2, 3 when the call is from CALCU and CALCV respectively). But careful perusal of the steps involved will assure the reader that the calculations are correct. The subsequent parts of the subroutine concern implementation of the TDMA technique: with the S-N traverse, assembling the TDMA coefficients, the calculation of the coefficients of the recurrence formula and the back-substitution to obtain the solution values of the S-N gridline.

3.9 Subroutine PRINT

The task of PRINT is to print out the values of a 2-D array PHI, together with associated heading HEAD, the last parameter of the call list. The X and Y coordinates are written horizontally and vertically around the output matrix of values. Again, as discussed in Section 3.8, the first two parameters of the call list, ISTART and JSTART, are usually 1,1 so that all internal and external values are printed to help diagnostics. Also, for U the XU values are printed and for V the YV values so that correct physical position values accompany the output of these variables which are located on displaced grids.

3.10 Subroutine PROMOD

The finite difference equation coefficients are formulated in each of the CALC subroutines on the assumption that each point is a usual internal point. Of course some points lie near or on the rectangular boundary of the flow domain in which the solution is sought and the general formulation of a CALC routine is not correct at these points. So that all required corrections to the general formulation are simple to understand and apply, each CALC routine calls a particular chapter of PROMOD (prob lem modifications). This subroutine has the task of modifying the values of the finite difference equation coefficients, or the variables, near walls or other boundaries where particular conditions apply. Each chapter concerns itself solely with one particular variable and each chapter considers in turn each of the boundaries around the solution domain. Boundary conditions are thus easy to formulate and simple to apply; and this novel feature makes PROMOD second only to MAIN as a subroutine to which a prospective user needs to apply his attention.

According to the value given to NCHAP (number of chapter required) of the call list, control proceeds to one and only one of the eight chapters of this subroutine. These chapters deal with:

C1 Properties

C2 U momentum

C3 V momentum

C4 Pressure correction

C5 Thermal Energy

C6 Turbulent kinetic energy

C7 Dissipation

C8 W momentum

where C5 is retained from the original TEACH arrangement, but is not used in the present version of the code.

Figure 5 reveals that amongst the boundaries of the solution domain only at the inlet is the usual formulation of the finite difference equations, as computed in the CALC subroutines, correct. At other boundaries of the solution domain appropriate sections of PROMOD have to supply the correct influence of the boundary conditions on the coupling coefficients a_j^ϕ (j = N, S, E, W and P) and components S_U^ϕ and S_p^ϕ of the linearized source term. In PROMOD each of the chapters is further divided into sections dealing with

(i) Northern top wall, I = 2 to NIM1

(ii) Southern symmetry axis, I = 2 to NIM1

(iii) Western side wall, J = 2 to NJM1

(iv) Eastern outlet, J = 2 to NJM1

In what follows all four cases of a N, S, E and W near-boundary point are dealt with simultaneously. One merely reads the appropriate line (designated by (N), (S), (E) and (W)) of the equations which follow, according to the particular case in which the reader is temporarily interested. Notation follows that of the boundary cells of Figs. 6 and 7.

Boundary conditions are inserted for each boundary cell. The first step generally involves breaking the link to any adjoining external cell by setting to zero the appropriate a_j of Eq. (11). Then, for each boundary

$$
\begin{aligned}
&AN\ (I,\ J) = 0 \quad (N)\\
&AS\ (I,\ J) = 0 \quad (S)\\
&AE\ (I,\ J) = 0 \quad (E)\\
&AW\ (I,\ J) = 0 \quad (W)
\end{aligned}
\tag{45}
$$

This step is followed by insertion of the correct boundary flux (diffusion and/or convection) for the cell boundary in question into the linearized source terms SU and SP, as exemplified in the following sections.

Neumann conditions. If zero normal gradient condition is the one to be applied, breaking the appropriate link is all that is required. For then the terms

$$
\begin{aligned}
&a_N\ (\phi_P - \phi_N) = 0 \quad (N)\\
&a_S\ (\phi_P - \phi_S) = 0 \quad (S)\\
&a_E\ (\phi_P - \phi_E) = 0 \quad (E)\\
&a_W\ (\phi_P - \phi_W) = 0 \quad (W)
\end{aligned}
\tag{46}
$$

are all zero in Eq. (11) so there is no flux through the boundary in question.

Dirichlet conditions. If the value of ϕ is prescribed as ϕ_B on the boundary then the usual boundary link is broken via Eq. (16). For a solid boundary the convective flux through the boundary face of the cell is zero, but the diffusive flux is typically nonzero. Hence the diffusion term of Eq. (8) is transferred to the right-hand side RHS of the equation where it is incorporated into SU(I,J) and SP(I,J); on the RHS it is expressed in terms of ϕ_P and ϕ_B as

$$\begin{aligned}
&\Gamma_B\ (\phi_B - \phi_P)\ r_B\ \Delta x/\delta y \qquad (N)\\
&\Gamma_B\ (\phi_B - \phi_P)\ r_B\ \Delta x/\delta y \qquad (S)\\
&\Gamma_B\ (\phi_B - \phi_P)\ r_B\ \Delta y/\delta x \qquad (E)\\
&\Gamma_B\ (\phi_B - \phi_P)\ r_B\ \Delta y/\delta x \qquad (W)
\end{aligned} \tag{47}$$

where Γ_B is the appropriate exchange coefficient for ϕ evaluated at point B. Appropriate cell boundary face areas appear here. North and south face areas are given by $r_B\ \Delta\theta\ \Delta x$ whereas east and west areas are expressed as $r_B\ \Delta\theta\ \Delta r$; but $\Delta\theta$ cancels from the governing equations. These expressions may be split between SU(I,J) and SP(I,J) as

$$SU(I,J) = SU\ (I,J) + \begin{cases} \Gamma_B\ r_B\ \phi_B\ \Delta x/\delta y & (N)\\ \Gamma_B\ r_B\ \phi_B\ \Delta x/\delta y & (S)\\ \Gamma_B\ r_B\ \phi_B\ \Delta y/\delta x & (E)\\ \Gamma_B\ r_B\ \phi_B\ \Delta y/\delta x & (W) \end{cases} \tag{48}$$

and

$$SP(I,J) = SP\ (I,J) - \begin{cases} \Gamma_B\ r_B\ \Delta x/\delta y & (N)\\ \Gamma_B\ r_B\ \Delta x/\delta y & (S)\\ \Gamma_B\ r_B\ \Delta y/\delta x & (E)\\ \Gamma_B\ r_B\ \Delta y/\delta x & (W) \end{cases} \tag{49}$$

Flux specified. If the flux of ϕ through a boundary face of the cell is specified. When brought to the RHS of Eq. (8) it may be approximately expressed as $b\phi_p + c$ where b and c are known (if possible, retaining (-b) and (c) both positive to aid the convergence of the finite difference iteration procedure). This is implemented by breaking the boundary link and augmenting the source terms appropriately:

$$SU\ (I,J) = SU\ (I,J) + c \quad (N,S,E,\text{ and } W) \tag{50}$$

$$SP\ (I,J) = SP\ (I,J) + b \quad (N,S,E,\text{ and } W) \tag{51}$$

Fixing a value at an internal point. The value of ϕ at a near-boundary point P may be fixed at a value of ϕ_F via setting

$$SU\ (I,J) = \phi_F\ 10^{30} \quad (N,S,E,\text{ and } W) \tag{52}$$

$$SP\ (I,J) = -10^{30} \quad (N,S,E,\text{ and } W) \tag{53}$$

so that these terms dominate in the equation for ϕ at P, with the solution $\phi_P = \phi_F$.

Convective and diffusive influences. If both convection and diffusion through a cell boundary occurs, the hybrid formulation should be used. In this situation the contribution C for the flux passing the surface in question (N,S,E, or W) is calculated according to Eq. (10) and used along with the other surface contributions and source terms to formulate the usual finite difference equation. This is ensured automatically in the program for a normal inflow/outflow boundary.

3.11 Subroutine STRMFN

Subroutine STRMFN has been developed to calculate the dimensionless stream function

$$STFN = \int_0^r urdr \Big/ \int_0^{d/2} urdr \tag{54}$$

for all internal field points of the u-cell grid. The quadrature formulation for calculating STFN is essentially the Trapezoidal Rule. The integration proceeds radially one step SNSV(J) at a time so that a finite difference mesh which is either uniform or nonuniform in the radial direction is permissible.

The coordinates of points constituting each of NSTLN streamlines with dimensionless stream function values STVAL ranging from 0.0 to 1.0 are subsequently calculated for plotting. The present code calculates the coordinates of eleven streamlines with ψ^* = 0.0, 0.1, ..., 1.0. Each streamline is represented by NI-1 points whose dimensionless x- and r-coordinates are stored as XUND(I) and YSTLND(I,K) (K = 1,..,11) respectively, where the k-index indicates the particular streamline. These two arrays are employed in producing COMPLOT streamline plots via the usual CALCOMP subroutines. Immediate line printer plots of alternate streamlines are obtained through the use of corresponding arrays XUDPLT(I) and YSLPLT(N,I) (N = 1,...,6) for the six streamlines ψ^* = 0.0, 0.2, ...,1.0.

3.12 Subroutine PLOT

The PLOT subroutine is a variant of the ϕ - profile line printer plot routine supplied with some versions of the GENMIX computer program. It is now described at length in a recent text.[34] In the present version supplied, it is called twice after each converged solution has been obtained. Firstly, with the parameter LARGE equal to 0 (giving a small-sized plot) and secondly with LARGE equal to 1 (giving a larger-sized plot). Prominent in the call list are XUDPLT(I) and YSLPLT(N,I) (N = 1,...,6) as described in Section 3.11 from which the required six streamlines are plotted. A 50 x 100 array is filled with alphanumeric characters as appropriate so as to give labels

0,2,4,6,8 and 1 corresponding to the nondimensional streamlines $\psi^* = 0.0$, $0.2, \ldots, 1.0$.

4. USER'S GUIDE

4.1 Problem Specification

General. The listing of the program, which is supplied in Appendix E, is used for precisely those calculations which have been described in Sections 2 and 3 of this report. The user of the program will wish to make other calculations; for this he must modify the program. Although the description of the program which was given in Section 3 will allow the discerning reader already to distinguish the parts of the program which must be altered, the following notes will aid him to make the modifications to suit his case, without going through the whole program.

The general principles which the user should adopt are:

(i) Read through MAIN, paying special attention to the Chapter titles, and other subtitles, and considering whether any of the features of the problem which relate to those titles have suffered changes which should be incorporated.

(ii) Make the corresponding changes to MAIN.

(iii) If modifications are desired to the turbulence model, introduce these in C2 of MAIN and/or C2 of PROPS.

(iv) If different wall boundary conditions are required, read carefully Sections 2.3 and 3.10 and then change PROMOD accordingly.

(v) Check that the COMMON statements are still adequate.

(vi) Leave the general parts of the program alone, including the CALC subroutines, INIT, PROPS, LISOLV and PRINT (except those minor nonstructural parts just mentioned).

The following discussion is organized under the headings: Input data specification, system specification, boundary conditions, and turbulence specification.

Input data specification. Much of the data is internally specified. However, if IREAD = .TRUE. has been declared in C0 of MAIN, the unformatted reading of all input data from disk storage is activated. This input requires the proper job control "card" to allocate the logical unit number of the READ statement to the disk storage data file from which the data is to be read. The dependent variable fields are read from disk storage as an improved (near-solution) initial estimate to reduce the number of iterations required for convergence.

As seen in Section 4.2, the X(I) and Y(J) [=R(J)] arrays preceed the dependent variable arrays sequentially written on each data file stored. Hence in C0 of MAIN, these two independent variable arrays are read to enable the next unformatted READ in C2 of MAIN to read the U(I,J), etc. dependent variables. After the solution is printed in C4 of MAIN, the dependent variable arrays are re-initialized for the next swirl loop by another series of unformatted reads from disk storage. Also, as discussed in Section 3, alphanumeric headings are read in C0 of MAIN using an A format field.

System specification. Minor variants of the sample grid system of Fig. 4 can be derived simply by appropriate modifications being made to the grid section of C1 of MAIN. Appropriate choice of the integers NI, NJ, ISTEP, JSTEP and JMAX(I) together with RLARGE (=D/2) ALTOT (= total length), EPSX (the gradual expansion rate in the axial direction) and Y(J) defines completely the axisymmetric geometry with INDCOS = 2. Assigning true or false to INCALU etc. activates the calls to subroutine CALCU etc. during the subsequent iteration cycle to update the U-velocity field. They also determine whether printing is required. Clearly, true settings will generally be the ones to use for all the dependent variables. Specification is also required for fluid properties, including Prandtl/Schmidt numbers and all material constants in SI units.

Boundary conditions. C1 of MAIN specifies boundary values of all the dependent variables. Hence specification of UIN etc. are required for the main inlet. Most inlet profiles of the turbulent flow are then taken to be uniform in C2 of MAIN and the user can easily amend this if desired. Employed here is the swirl number block SWNB or vane angle block VANB according to the value of NSBR (number for solid body rotation). The specification NSBR = 1 activates SWNB, which gives solid body rotation from a swirl generator, and NSBR = 0 activates VANB for a flat swirl velocity profile from swirl vanes. In either case, the desired inlet values of swirl velocity W are assigned appropriately.

Initial field values of all the dependent variables are specified in C2 of INIT and C2 of MAIN. The user will be content to leave these alone, for as soon as the iteration process is underway, these estimates are improved. Only occasionally in some cases will they be too inaccurate to permit a smooth iterative process to continue from their values - in such cases more realistic values closer to the true solution and/or underrelaxation must be specified.

Turbulence specification. Minor adjustments to the turbulence model already in the program can be simply achieved by altering the turbulence constants part of C1 of MAIN and C2 of PROPS for the viscosity calculation. If one desired to make a computation for laminar flow, VIS must be filled with appropriate values in the DO 100 loop of PROPS. Also, the logical variables INCALK and INCALD could be specified as false to suppress the calculation of k and ε and the complexity in PROMOD for velocities near wall boundaries would be removed. When a different turbulence model is required, of course, C2 of PROPS must reflect this. Should a nonisotropic or direct turbulent stress specification model be required, more far-reaching changes would be required to the diffusion term calculations in all the CALC subroutines, possibly with other CALC routines for solving stress equations. It is here that current knowledge is uncertain and much current research activity is being directed toward more realistic turbulence models.

4.2 Iteration and Accuracy

Iteration control. The iteration process is monitored by comparing the sum of the absolute values of residual source of mass in the filed, RESORM, with a preset value, SORMAX, representing maximum source. Similarly for other variables U, V, W, etc. iteration is terminated for a particular problem when the largest residual source is less than SORMAX, or number of iterations NITER is greater than the maximum number of iterations MAXIT to be allowed. By analogy a divergence criterion can also be included if required, so as to prematurely terminate a computation giving results which appear not to be converging.

Fortran variables which are used to influence the iteration behavior are NSWPU etc. (representing number of update sweeps for U etc.) and URFU etc, (representing underrelaxation for U etc.). If divergence is found, the remedy often lies in increasing the former (especially NSWPP for pressure) and reducing the latter. However, the accompanying listing includes reasonable values which have been found convenient to use. They are automatically re-assigned in the iteration process according to the inlet swirl strength and number of iterations made toward convergence, as inspection of the listing reveals.

Output. At every iteration, RESORM etc. are printed out along with monitored values of variables at the location specified by IMON and JMON. Full field prints presently after 100 iterations and increments of 50 thereafter. Full field prints are obtained for each dependent variable with a true logical variable (like INCALU for U-veloctiy) by way of a call to subroutine PRINT. The user, of course, is entirely free to make any output he wishes at each iteration stage in C3 of MAIN and to make a field print of any 2-D array via a call to PRINT. This structure is a flexible one which can be utilized to advantage.

If IWRITE = .TRUE. has been specified in CO of MAIN, the unformatted writing of all output data onto disk storage is activated. This output requires the proper job control "card" to allocate the logical unit number of the WRITE statement to the disk storage data file which is to receive the data. First, the X(I) and then the Y(J) [=R(J)] arrays are written to disk storage allocated to logical unit 11, as seen in the initial output portion of MAIN, C2. Also, located here is the writing of XUND(I), nondimensional XU(I), to disk storage allocated to logical unit 14.

Upon completion of the solution process for each swirl loop each field variable is nondimensionalized and printed (if NONDIM = .TRUE.) subsequent to the printing of dimensional variables. Then, the following dependent variable arrays are written onto disk (logical unit 11): U(I,J), V(I,J), W(I,J), P(I,J), TE(I,J), ED(I,J), VIS(I,J) and STFN(I,J). Immediately following is the WRITE of YSTLND(I,K) to disk (logical unit 14), where YSTLND(I,K) contains the dimensionless radial coordinate corresponding to each dimensionless axial coordinate XUND(I) along each K streamline. These arrays are subsequently used in obtaining CALCOMP streamline plots.

Independence of grid size. A computation may be considered to be accurate if alterations in the grid size produce no significant changes in the values of dependent variables, or the fluxes, at the points in the flow or its boundary which are interesting to the user of the program. Such alterations are obtained by changing the values of NI and NJ (number of grid intervals in I and J (i.e. X and Y) directions) along with EPSX and Y(J). Specification of these quantities is made in C1 of MAIN. Of course, large values of NI and NJ increase the computer time significantly, so smallest values should be used that are consistent with acceptable accuracy. There is no way but trial to establish what these values are, but a 23 x 21 or 35 x 21 grid is probably sufficient, as available by simple choice in the delivered code.

Comparison with exact or experimental values. Further tests of accuracy can be made by setting the geometry and boundary conditions of the program to correspond to a problem for which exact solutions or experimental results are available. For example, the inlet step could be dispensed with and standard pipe flow results. In either case the presence of swirl and turbulence is optional. Hence comparison of the predictions of the program for this degenerate case with exact or experimental results can provide valuable insight into the accuracy with which the differential equations are being solved.

'Trouble-shooting' - It is possible to mention only a few of the remedies which can be employed when the program, in the course of being adapted to a new problem, generates computations that are evidently faulty.

To prevent divergence, the source terms should be formulated so that -SP is always positive for each ϕ variable. This keeps the point P dominant in its finite difference form with its N, S, E and W neighbors and aids the stability of the iteration process. The forms shown in Table 2 enjoy this characteristic.

Lack of conservation of properties which ought to be conserved is often the result of incompatible specification of boundary conditions and fluid properties, for example. One has to take extreme care. It is not possible to say much more in general than that inadvertently introduced incompatibilities are among the most common causes of error.

4.3 The Sample Computation

The sample computation code listing (Appendix E) and the results given in the microfiche supplement are concerned with the inert turbulent flow in an idealized combustion chamber. An air flow (constant density $\rho = 1.211$ kg/m^3 and molecular viscosity $\mu = 1.8 \times 10^{-5}$ kg/ms) enters a 45 deg. expansion ($\alpha =$ 45 deg.) from an inlet pipe (radius d/2 = 0.03125 m) to a large pipe (radius D/2 = 0.0625 m). The total length of the flow domain is 0.5 m so as to be sufficient to enclose any expected recirculation zones. Computer runs through a range of seven inlet swirl vane angles ϕ(= arctan w_{in}/u_{in}) equal to 0, 45,55, 60, 65, 68 and 70 deg. are given, although serious production runs with the larger swirl strengths are made with the total length further extended to about 0.8 m in order to cover a larger domain associated with a longer central recirculation zone. A maximum of 200 iterations for each swirl strength is allowed, with the solutions for each value of ϕ being used as the initial starting values for the next higher value of ϕ.

The flow domain is covered by a 23 x 21 mesh system (NI = 23 for the x-direction, NJ = 21 for the r-direction). The sidewall angle of α = 45 deg. is accommodated by way of four steps each of three cells high in the radial direction. The specification is

ISTEP = 2

JSTEP = 8

JMAX(1) = 8

JMAX(2) = 11

JMAX(3) = 14

JMAX(4) = 17

JMAX(I) = 20 (I = 5,6,...,21)

In the x-direction a uniformly expanding grid up to x = 0.375 m at I = 20 is specified via EPSX = 1.11, then equidistant addition grid lines are specified up to x = 0.5 m at NI = 23. This grid is appropriate for side-wall angles α = 45 and 90 deg., though with α = 70 deg. a uniformly expanding grid up to x = 0.375 m at I = 30 is specified via EPSX = 1.102, then equidistant additional grid lines are specified up to x = 0.55 m at NI = 35. In the r-direction the 21 grid lines are clustered near the expansion corner, centerline and north wall since additional refinement is required in regions with expected large gradients. The choice of x-grid is determined by setting IFINE to be true or false, giving fine or coarse grid system, respectively. Both grids are arranged

(i) to give good resolution in strategic locations near the inlet, exit, projecting corners and front and back stagnation points,

(ii) to give corresponding axial locations at which to compare predictions with present and previous measurements.

Finally, the choice of logical variables in Chapter 0 of MAIN is self-explanatory giving rather detailed output but not reading from or writing to any disk storage locations.

When the user wishes to compute a problem with a sidewall angle α different from 45 deg. [as in the listing supplied in Appendix E] he has to specify the x and r [$\equiv$ y] grid line locations and JMAX(I) for each I. Many variants of the flow domain of Fig. 5 discussed in Section 2.3 can thus be investigated. Auxiliary computations lead to the required specification. For example, other values of α are obtained by modifying the x-grid spacing near the inlet, which depends on NI, EPSX and ALTOT, and comparing it with the corresponding r-grid spacing near the expansion corner. That is, XU(3) - XU(2) is compared with YV(JSTEP+2) - YV(JSTEP+1) to get the aspect ratio by the expansion corner. When a multi-cell step is envisioned a greater range of YV values are considered. Corresponding values of α can be tabulated with various combinations of NI and EPSX, and the user can then judiciously choose an appropriate combination with JMAX(I) values for his needs.

Another parameter frequently requiring different values is swirl strength. The program is set up via LFS and LFSMAX to sequentially calculate through a range of seven swirl vane angles. These angles are specified in a DATA statement for the array VANB which is activated if NSBR = 0. If NSBR = 1 the swirl number for consecutive swirl numbers to be investigated is specified via a DATA statement for the array SWNB.

5. CLOSURE

The STARPIC computer program has been developed to predict swirling recirculating inert turbulent flows in axisymmetric combustion chambers. The technique involves a staggered grid system for axial and radial velocities, a line relaxation procedure for efficient solution of the equations, a two-equation $k - \varepsilon$ turbulence model, a stairstep boundary representation of the expansion flow, and realistic accommodation of swirl effects. Predictions of this type allow some results to be obtained more cheaply, quickly, and correctly than currently possible by the almost exclusive use of experimental means. Further development and application is providing a valuable supplementary technique for designers of practical combustion equipment.

This report has dealt with the computational problem and shown how the mathematical basis and computational scheme may be translated into a computer program. A flow chart, Fortran 4 listing, notes about various subroutines and a user's guide are supplied as an aid to prospective users of the code.

6. REFERENCES

1. Lilley, D. G., Flowfield Modeling in Pratical Combustors: A Review, Journal of Energy, Vol. 3, July-Aug. 1979, pp. 193-210.
2. Lilley, D. G., Prospects for Computer Modeling in Ramjet Combustors, AIAA Paper No. 80-1189, Hartford, CT, June 30-July 2, 1980.
3. Syred, N. and Beer, J. M., Combustion in Swirling Flows: A Review, Combustion and Flame, Vol. 23, 1974, pp. 143-201.
4. Lilley, D. G., Swirl Flows in Combustion: A Review, AIAA Journal, Vol. 15, No. 8, August 1977, pp. 1063-1078.
5. Beer, J. M. and Chigier, N. A., Combustion Aerodynamics, Applied Science, London and Halstead-Wiley, New York, 1972.
6. Gupta, A. K., Lilley, D. G. and Syred, N., Swirl Flows, Abacus Press, Tunbridge Wells, England, 1981.
7. Rhode, D. L., Lilley, D. G. and McLaughlin, D. K., On the Prediction of Swirling Flowfields Found in Axisymmetric Combustor Geometries. ASME Symposium on Fluid Mechanics of Combustion Systems, held in Boulder, Colorado, June 22-24, 1981.
8. Gerstein, M. (ed), Fundamentals of Gas Turbine Combustion, NASA-CP-2087, 1979. Workshop held at NASA Lewis Research Center, Cleveland, OH, Feb. 6-7, 1979.
9. Murthy, S. N. B., Summary Report DOD Colloquium on Gas Turbine Combustor Modeling. Project SQUID Technical Report PU-R1-79, School of Mech. Engr., Purdue University [for ONR], Nov. 1979.
10. Lefebvre, A. H. (ed), Gas Turbine Combustor Design Problems, Hemisphere-McGraw-Hill, New York, 1980.
11. Launder, B. E. and Spalding, D. B., Mathematical Models of Turbulence, Academic Press, London, England, 1972.

12. Launder, B. E. and Spalding, D. B., The Numerical Computation of Turbulent Flows, Comp. Methods in Appl. Mech. and Engrg., Vol. 3, March 1974, pp. 269-289.

13. Roache, P. J., Computational Fluid Dynamics, Hermosa, Albuquerque, NM, 1972.

14. Chow, C.-Y., An Introduction to Computational Fluid Mechanics, Wiley, New York, NY, 1979.

15. Patankar, S. V., Numerical Heat Transfer and Fluid Flow, Hemisphere-McGraw-Hill, New York, 1980.

16. Gosman, A. D., Pun, W. M., Runchal, A. K., Spalding, D. B., and Wolfshtein, M. W., Heat and Mass Transfer in Recirculating Flows, Academic Press, London, England, 1969.

17. Gosman, A. D., and Pun, W. M., Calculation of Recirculating Flows, Rept. No. HTS/74/12, 1974, Dept. of Mechanical Engineering, Imperial College, London, England.

18. Gosman, A. D. and Ideriah, F. J. K., TEACH-2E: A General Computer Program for Two-Dimensional, Turbulent, Recirculating Flows. Report, Dept. of Mech. Eng., Imperial College, London, June 1976.

19. Lilley, D. G., Primitive Pressure-Velocity Code for the Computation of Strongly Swirling Flows, AIAA Journal, Vol. 14, June 1976, pp. 749-756.

20. Khalil, E. E., Spalding, D. B. and Whitelaw, J. H., The Calculation of Local Flow Properties in Two-Dimensional Furnaces, Int. J. Heat Mass Trans., Vol. 18, 1975, pp. 775-791.

21. Novick, A. S., Miles, G. A., and Lilley, D. G., Numerical Simulation of Combustor Flowfields: A Primitive Variable Design Capability, J. of Energy, Vol. 3, No. 2, March-April, 1979, pp. 95-105.

22. Wuerer, J. E. and Samuelsen, G. S., Predictive Modeling of Backmixed Combustor Flows: Mass and Momentum Transport, AIAA Paper 79-0215, New Orleans, LA., Jan. 15-17, 1979.

23. Habib, M. A. and Whitelaw, J. H., Velocity Characteristics of Confined Coaxial Jets With and Without Swirl, ASME Paper No. 79-WA/FE-21, New York, Dec. 2-7, 1979.

24. El Banhawy, Y. and Whitelaw, J. H., Calculation of the Flow Properties of a Confined Kerosene-Spray Flame, AIAA J., Vil. 18, No. 2, Dec. 1980, pp. 1503-1510.

25. Srinivasan, R. and Mongia, H. C., Numerical Computations of Swirling Recirculating Flows, Final Report, NASA-CR-165196, Sept. 1980.

26. Patankar, S. V. and Spalding, D. B., A Computer Model for Three-Demensional Flow in Furnaces, 14th Symp. (Int.) on Comb., The Comb. Inst., Pittsburgh, Pa., 1973, pp. 605-614.

27. Serag-Eldin, M. A. and Spalding, D. B., Computations of Three-Dimensional Gas-Turbine Combustion Chamber Flows, Trans. ASME J. of Eng. for Power, Vol. 101, July 1979, pp. 326-336.

28. Mongia, H. C. and Reynolds, R. S., Combustor Design Criteria Validation Vol. III - User's Manual, Report USARTL-TR-78-55C, U.S. Army Res and Tech. Lab., Ft. Eustis, VA, Feb. 1979. [See also Vols. I and II].

29. Swithenbank, J., Turan, A. and Felton, P. G., Three-Dimensional Two-Phase Mathematical Modeling of Gas Turbine Combustors, in Gas Turbine Combustor Design Problems (Lefebvre, A. H. ed.), Hemisphere-McGraw-Hill, New York, 1980, pp. 249-314.

30. Spalding, D. B., A Novel Finite Difference Formulation of Differential Expressions Involving Both First and Second Derivatives, Int. J. Num. Meth. in Eng., Vol. 4, 1972, pp. 551-559.

31. Weske, D. R. and Sturov, G. Ye, Experimental Study of Turbulent Swirled Flows in a Cylindrical Tube, Fluid Mechanics - Soviets Research, Vol. 3, No. 1, Jan.-Feb., 1974, pp. 77-82.

32. Blackshall, R. G. and Landis, F., The Boundary-Layer Velocity Distribution in Turbulent Swirling Pipe Flow, Journal Basic Engr., ASME Trans., Dec. 1969, pp. 728-733.

33. Date, A. W., Prediction of Fully-Developed Flow in a Tube Containing A Twisted Tape, Int. Jl. Heat Mass Transfer, Vol. 17, pp. 845-859.

34. Spalding, D. B., GENMIX - A General Computer Program for Two-Dimensional Parabolic Phenomena. Pergamon Press, Oxford, England, 1977.

7. NOMENCLATURE

A	Cell face area, Eq. (8)
a	Coupling coefficient, Eq. (11)
C	Contribution to cell surface integral, Eq. (10)
C,U,V	Control cell volumes for ϕ, u, v, Fig. 3
D	Chamber diameter, Fig. 1
d	Nozzle diameter, Fig. 1
E	Constant in law of wall, Eq. (24)
f	Underrelaxation parameter, Eq. (42)
G	k - generation term, Eq. (4)
I,J	Mesh point, Fig. 2
k	Kinetic energy of turbulence, Eq. (1)
$\dot{m}_{net}$	Net outflow of mass from cell, Eq. (9)
Pe	Cell Peclet number = $\rho u\delta/\Gamma$, Eq. (10)
p	Time-mean pressure, Eq. (1)
R	Residual source, Eq. (44)
S	Source term (with subscript), Eq. (1)
S_P, S_U	Components of linearized source term, Eq. (8)
V	Magnitude of total velocity vector, Eq. (24)
$\underline{v} = (u,v,w)$	Time-mean velocity (in x,r,θ directions), Eq. (1)
x,r,θ	Axial, radial, circumferential polar co-ordinates, Eq. (1)
y	Distance normal to a wall, Eq. (25)
Γ	Turbulent exchange coefficient, Eq. (1)
α	Side-wall angle, Fig. 1
Γ	Exchange coefficient, Eq. (7)

δx	Axial distance between two neighboring mesh points, Fig. 4
ε	Turbulence energy dissipation rate, Eq. (1)
κ	Constant in log law, Eq. (24)
μ	Effective viscosity, Eq. (5)
ρ	Time-mean density, Eq. (1)
σ	Prandtl-Schmidt number, Eq. (6)
τ	Wall shear stress, Eq. (25)
ϕ	Swirl vane angle, Fig. 1 dependent variable, Eq. (1)
ψ	Stream function, Eq. (7)

<u>Superscripts</u>

old	Last iterate value, Eq. (9)
*	Preliminary u, v and p field based on estimated pressure field p*, Eqs. (12)-(15)
$'$	Correction value to u*,v*,p* to get u,v,p, Eqs. (12)-(15)

<u>Subscripts</u>

in	Inlet, Fig. 1
ℓ	Laminar value, Eq. (5)
n,s,e,w	North, south, east, west faces of cell, Eq. (8)
P,N,S,E,W	Point, north, south, east, west neighbors, Fig. 2
t	Total, Eq. (25)

APPENDIX A DERIVATION OF THE HYBRID DIFFERENCING SCHEME

In the discussion of Section 2.2, it was remarked that a hybrid differencing scheme is used in representing the combined effects of convection and diffusion at cell boundaries. It possesses certain advantages over the well-known central and upstream differencing schemes, and it is convenient to provide here sufficient derivation, in the context of the present 2-D axisymmetric flow problem. The reasons and choices are aided by comparison of possible finite difference representations with the exact solution of a simplified 1-D convection and diffusion problem, whose solution values and flux calculations therefrom provides useful guidance.

Consider the transport across one face of a control volume, for example the western face of area A_w normal to the x-direction which lies midway between the gridpoints W and P distant δx apart, as shown earlier in Fig. 4. The contribution C_w of the convection and diffusion terms of Eq. (1) to the surface integral for the western face is

$$C_w = [\rho u \phi - \Gamma_\phi \frac{\partial \phi}{\partial x}]_w A_w \tag{A1}$$

and the task is to derive a finite difference expression to represent the right hand side realistically in terms of ϕ and Γ_ϕ values at W and P using appropriate differencing. The choice will be aided by comparison with the exact solution of

$$\rho u \phi - \Gamma_\phi \frac{\partial \phi}{\partial x} = 0 \tag{A2}$$

in the interval $W \leq x \leq P$, taken without loss of generality to be $0 \leq x \leq \delta x$, where $\Gamma_\phi = (\Gamma_\phi)_w$ = constant and ρ and u are presumed to be constant known values evaluated at $x = \delta x/2$. Solving this one-dimensional

Eq. (A2) subject to the two grid values

$$\begin{aligned} x &= 0 \qquad & \phi &= \phi_W \\ x &= \delta x \qquad & \phi &= \phi_P \end{aligned} \tag{A3}$$

yields the solution

$$\phi = \phi_W + (\phi_P - \phi_W)\left[\frac{e^{Pe\ x/\delta x} - 1}{e^{Pe} - 1}\right] \tag{A4}$$

where the cell Peclet number has been defined by

$$Pe = \rho u\ \delta x\ /\ \Gamma_\phi \tag{A5}$$

which is evaluated at $x = \delta x/2$, i.e. at the western face of the cell around P. The exact solution, Eq. (A4), is portrayed in Fig. 12 where inspection shows that the value of ϕ always lies within the extreme values of ϕ_W and ϕ_P. When Pe becomes very large and positive, because of axial velocity u being very large and from left to right in Fig. 3, the ϕ-x curve is nearly horizontal and nearly equal to ϕ_W at all points, with a steep rise to the value Φ_P as x approaches δx (i.e. near P). Conversely when Pe is very large and negative, because of axial velocity u being very large and from right to left for example, the ϕ-x curve is nearly equal to ϕ_P at all points with the steep part near x = 0 (i.e. near W). When Pe is near to zero in value, the linear interpolation straight line is close to the exact solution.

The present task is to look at the net transfer of ϕ across the mid-plane at $x = \delta x/2$ and to deduce a satisfactory finite difference analogue of the differential equation Eq. (A2) in terms of ϕ_W and ϕ_P. The flux of ϕ across the mid-plane at $x = \delta x/2$ is

$$Q_{\delta x/2} = \left[\rho u \phi - \Gamma_\phi \frac{\partial \phi}{\partial x}\right]_{x = \delta x/2} \tag{A6}$$

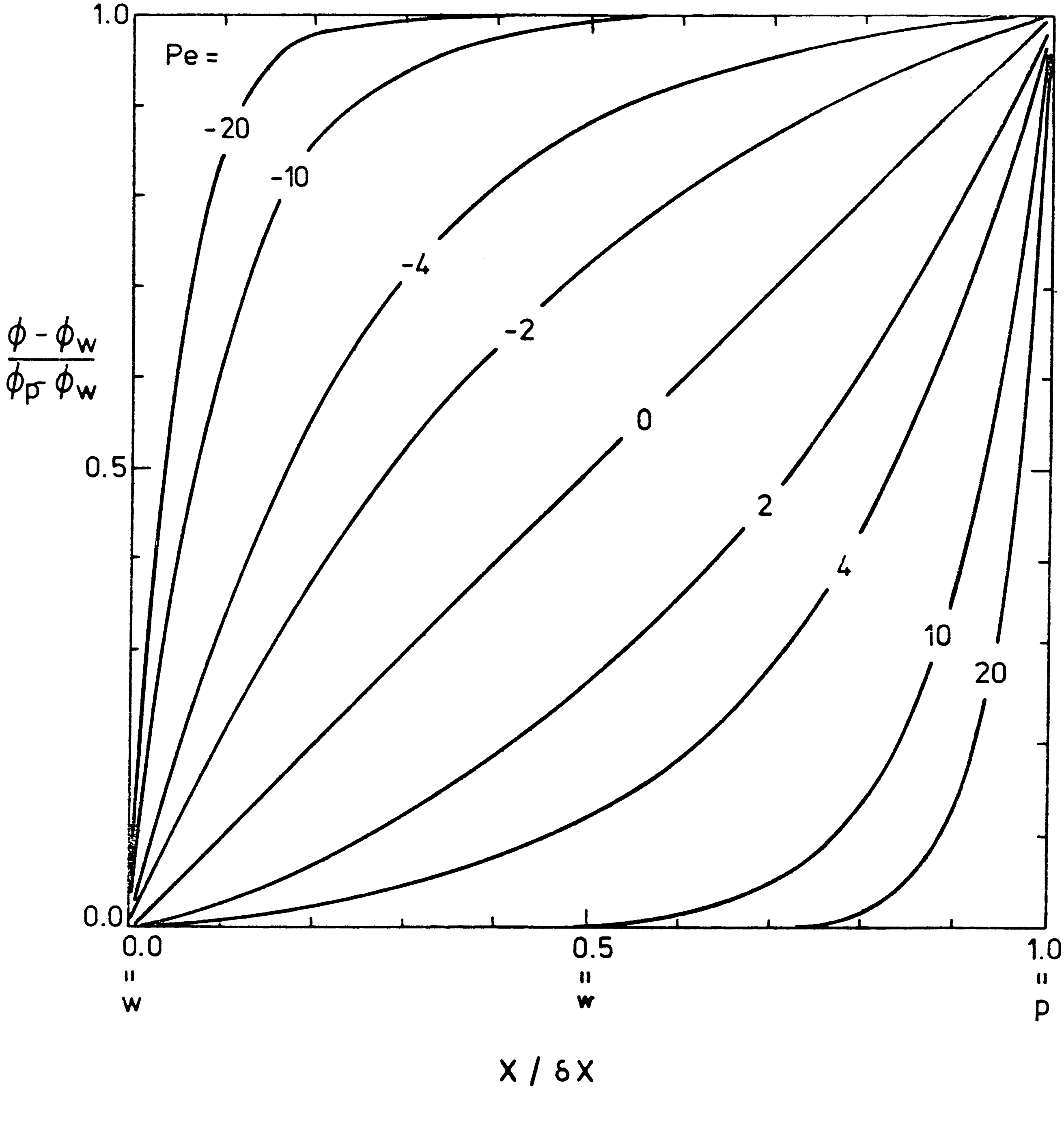


Fig. 12. ϕ versus x curves for different values of the cell Peclet number Pe (assuming $\phi_W < \phi_p$).

and upon substituting the exact solution for ϕ and $\frac{\partial \phi}{\partial x}$ at $x = \delta x/2$ using Eq. (A4) yields (see Appendix B)

$$Q_{\delta x/2} = (\rho u)_{\delta x/2} \left\{ \frac{\phi_W e^{Pe} - \phi_P}{e^{Pe} - 1} \right\}$$
$$= (\rho u)_{\delta x/2} \left\{ \frac{\phi_W e^{Pe}}{e^{Pe} - 1} - \frac{\phi_P e^{-Pe}}{1 - e^{-Pe}} \right\} \tag{A7}$$

Three limits of this expression can be examined:

$$Pe \rightarrow +\infty \qquad Q_{\delta x/2} \rightarrow (\rho u)_{\delta x/2} \cdot \phi_W \tag{A8}$$

$$Pe \rightarrow -\infty \qquad Q_{\delta x/2} \rightarrow (\rho u)_{\delta x/2} \cdot \phi_P \tag{A9}$$

$$|Pe| \rightarrow 0 \qquad Q_{\delta x/2} \rightarrow (\rho u)_{\delta x/2} \left\{ \phi_W \frac{1 + Pe + ---}{Pe\ (1 + Pe/2 + ---)} - \phi_P \frac{1 - Pe + ---}{Pe\ (1 - Pe/2 + ---)} \right\} \tag{A10}$$

$$= (\rho u)_{\delta x/2} (\phi_W + \phi_P)/2 - \Gamma_{\delta x/2} \frac{\phi_P - \phi_W}{\delta x} \tag{A11}$$

where terms of order Pe^2 and higher have been omitted from the last expression. Rearrangement and substitution of $\Gamma_{\phi,\delta x/2} / \delta x$ for $(\rho u)_{\delta x/2} / Pe$ has been made to obtain the final form of Eq. (A11). This last equation is equivalent to central differencing of the diffusion part (in square brackets) of Eq. (A1). The first two limits of Eqs. (A8) and (A9) are the upstream convection dominated limits where diffusion terms are negligible. In computing it is necessary to use a finite difference expression for the flux passing the surface for the western face contribution

$$C_W = Q_{\delta x/2} \cdot A_W \tag{A12}$$

which exhibits these three limiting cases. The hybrid scheme recommended and used in Refs. 17-23 exhibits these properties via the use of

$$C_w = \begin{cases} (\rho u)_w \, A_w \, (\phi_W + \phi_P) \, /2 - (\Gamma_\phi)_w \, A_w(\phi_P - \phi_W)/\delta x & \text{for } |Pe| < 2 \\ (\rho u)_w \, A_w \, \phi_W & \text{for } Pe \geq 2 \\ (\rho u)_w \, A_w \, \phi_P & \text{for } Pe \leq -2 \end{cases} \tag{A13}$$

where $Pe = (\rho u)_w \, \delta x/(\Gamma_\phi)_w$ is the cell Peclet number calculated at w. This hybrid scheme is denoted by HS.

Figure 13 shows the nondimensional flux passing the mid-plane at $x - \delta x/2$ versus Peclet number Pe. Four lines are shown representing the exact solution ES, the hybrid scheme HS, the central difference scheme CDS and the upwind difference scheme UDS. These lines show the overall superiority of the HS over the CDS and UDS approaches, for the particular case considered in Appendix C. However, the result is quite general.[30] These different schemes and the superiority of the hybrid scheme recommended and used in this work[7] are fully discussed in Appendix C.

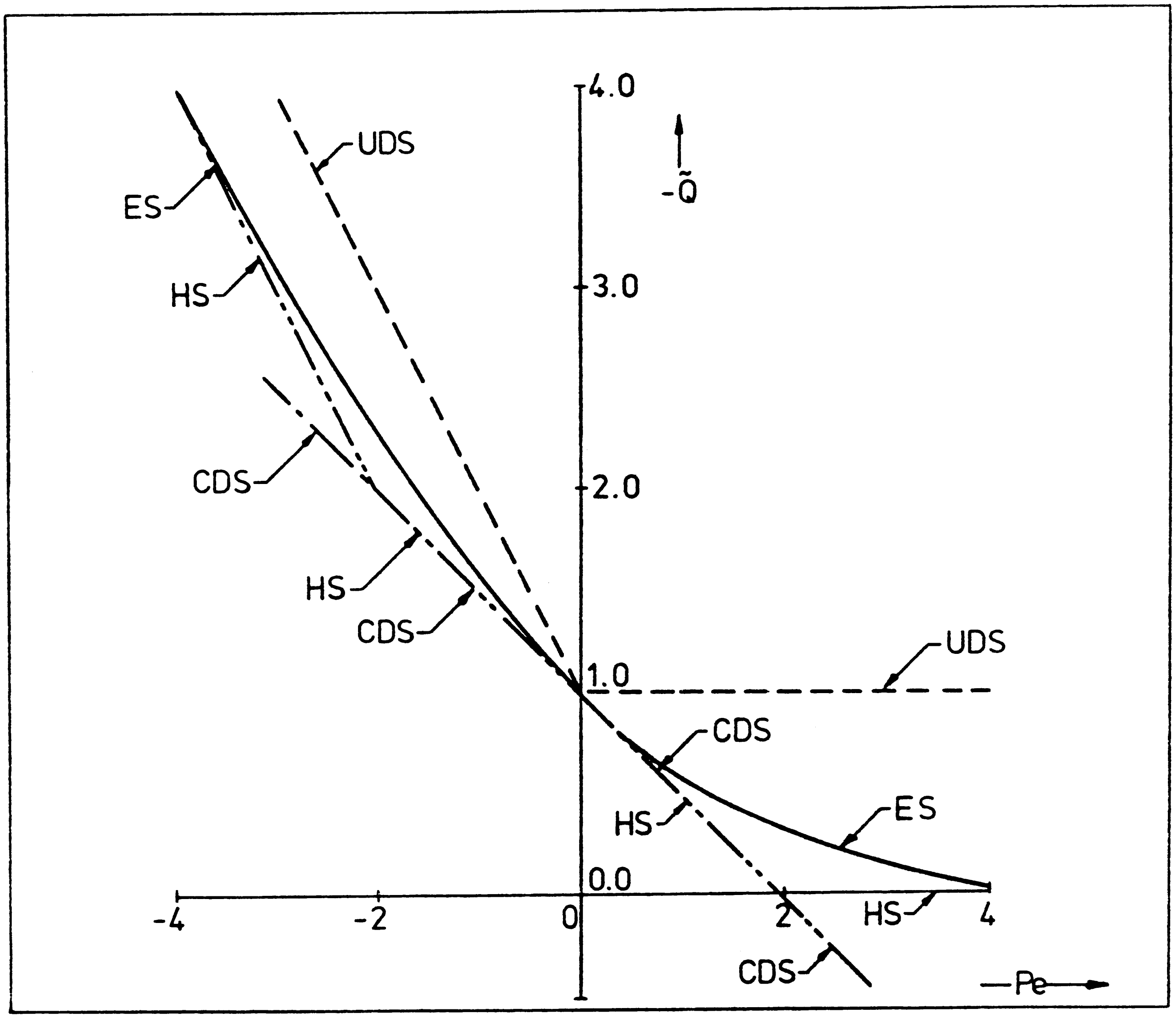


Fig. 13. Comparison of exact flux passing through western face with other approaches, as a function of Peclet number (ES, CDS, UDS, AND HS are described in Appendix C).

APPENDIX B DERIVATION OF EQ. (A7) FROM EQ. (A6)

Equation (A4) gives ϕ and $\frac{\partial\phi}{\partial x}$ evaluated at $x = \partial x/2$ as

$$\phi_{\delta x/2} = \phi_W + (\phi_P - \phi_W)\,\frac{e^{Pe/2} - 1}{e^{Pe} - 1} \tag{B1}$$

and

$$\left.\frac{\partial\phi}{\partial x}\right|_{\delta x/2} = (\phi_P - \phi_W)\cdot\frac{1}{e^{Pe} - 1}\cdot\frac{Pe}{\delta x}\cdot e^{Pe/2} \tag{B2}$$

Substitution into Eq. (A6) gives, where $\rho u = (\rho u)_{\delta x/2}$:

$$\begin{aligned}
Q_{\delta x/2} &= \rho u\left\{\phi_W + (\phi_P - \phi_W)\,\frac{e^{Pe/2} - 1}{e^{Pe} - 1}\right\} \\
&\quad - \Gamma_\phi\left\{(\phi_P - \phi_W)\,\frac{1}{e^{Pe} - 1}\cdot\frac{Pe}{\delta x}\cdot e^{Pe/2}\right\} \\
&= \rho u\,\phi_W\left\{1 - \frac{e^{Pe/2} - 1}{e^{Pe} - 1} + \frac{e^{Pe/2}}{e^{Pe} - 1}\right\} \\
&\quad + \rho u\,\phi_P\left\{\frac{e^{Pe/2} - 1}{e^{Pe} - 1} - \frac{e^{Pe/2}}{e^{Pe} - 1}\right\}
\end{aligned} \tag{B3}$$

on substituting for Γ_ϕ in terms of the Peclet number $Pe = \rho u\delta x/\Gamma_\phi$.

continuing

$$\begin{aligned}
Q_{\delta x/2} &= \rho u\,\phi_W\left\{\frac{e^{Pe} - 1 + 1}{e^{Pe} - 1}\right\} + \rho u\,\phi_P\left\{\frac{-1}{e^{Pe} - 1}\right\} \\
&= (\rho u)_{\delta x/2}\left\{\phi_W\frac{e^{Pe}}{e^{Pe} - 1} - \phi_P\,\frac{1}{e^{Pe} - 1}\right\}
\end{aligned} \tag{B4}$$

which is equivalent to Eq. (A7).

APPENDIX C THE SUPERIORITY OF THE HYBRID DIFFERENCING SCHEME

Here the concern is with exhibiting possible finite difference analogues of Eq. (A7) which expresses the exact solution for the flux $Q = Q_{\delta x/2}$ passing the mid-plane $x = \delta x/2$, as a function of local density and axial velocity evaluated at this point $\rho u = (\rho u)_{\delta x/2}$, the local Peclet number Pe and the ϕ-values at W and P. Three limiting cases for $Pe \to +\infty$, $Pe \to -\infty$ and $|Pe| \to 0$ were given in Eqs. (A8) - (A11), the first two being upstream convection dominated and the last one being equivalent to simple averaging for the convection term and central differencing for the diffusion term.

In computing it is advantageous to use a finite difference analogue of Eq. (A2) which gives the best fit for this flux Q over a range of ρ, u, Γ and δx likely to be encountered, as expressed via the Peclet number Pe. The suffix ϕ on Γ is omitted from here onward to avoid confusion. Nonsubscripted variables are evaluated at $x = \delta x/2$. The three popular practices, and the exact solution, are:

(i) CDS - the central difference scheme. For all Pe numbers, use

$$Q = \rho u\,(\phi_P + \phi_W)/2 - \Gamma(\phi_P - \phi_W)/\delta x \tag{C1}$$

(ii) UDS - the upstream difference scheme. For all Pe numbers, use

$$Q = \begin{cases} \rho u \phi_W - \Gamma(\phi_P - \phi_W)/\delta x & (u > 0) \\ \rho u \phi_P - \Gamma(\phi_P - \phi_W)/\delta x & (u < 0) \end{cases} \tag{C2}$$

(iii) HS - the hybrid scheme. According to Pe and u, use (see Eq. (A13)):

$$Q = \begin{cases} \rho u\,(\phi_P + \phi_W)/2 - \Gamma(\phi_P - \phi_W)/\delta x & (|Pe| < 2) \\ \rho u\,\phi_W & (Pe \geq 2,\ u > 0) \\ \rho u\,\phi_P & (Pe \leq -2,\ u < 0) \end{cases} \tag{C3}$$

(iv) ES - the exact solution. According to Pe number, use (see Eq. (A7)):

$$Q = \rho u \left\{ \frac{\phi_W \; e^{Pe}}{e^{Pe}-1} - \phi_P \frac{e^{-Pe}}{1 - e^{-Pe}} \right\} \tag{C4}$$

We wish to determine which of the CDS, UDS or HS is generally the best to use as an approximation to ES, over a range of Pe numbers. Nondimensionalizing the fluxes via

$$\hat{Q} = Q \cdot \frac{\delta x}{\Gamma} \cdot \frac{1}{\phi_P - \phi_W} \tag{C5}$$

gives the following forms (using Pe = $\rho u \delta x / \Gamma_\phi$ when required):

(i) CDS
$$\hat{Q} = \tfrac{1}{2} \cdot Pe \cdot \frac{\phi_P + \phi_W}{\phi_P - \phi_W} - 1 \tag{C6}$$

(ii) UDS
$$\hat{Q} = \begin{cases} Pe \cdot \dfrac{\phi_W}{\phi_P - \phi_W} - 1 & (u > 0) \\ Pe \cdot \dfrac{\phi_P}{\phi_P - \phi_W} - 1 & (u < 0) \end{cases} \tag{C7}$$

(iii) HS
$$\hat{Q} = \begin{cases} \tfrac{1}{2} \cdot Pe. \dfrac{\phi_P + \phi_W}{\phi_P - \phi_W} - 1 & (|Pe| < 2) \\ Pe \cdot \dfrac{\phi_W}{\phi_P - \phi_W} & (Pe \geq 2,\ u > 0) \\ Pe \cdot \dfrac{\phi_P}{\phi_P - \phi_W} & (Pe \leq 2,\ u < 0) \end{cases} \tag{C8}$$

(iv) ES
$$\hat{Q} = Pe \left\{ \frac{\phi_W \; e^{Pe}}{e^{Pe} - 1} - \frac{\phi_P \; e^{-Pe}}{1 - e^{-Pe}} \right\} \tag{C9}$$

It is useful to examine limiting cases of the ES, in a manner similar to that used in obtaining Eqs. (A8) - (A11). Now they are

$$\hat{Q} \rightarrow \begin{cases} -1 & (|PE| \rightarrow 0) \\ Pe \cdot \dfrac{\phi_W}{\phi_P - \phi_W} & (Pe \rightarrow +\infty) \\ Pe \cdot \dfrac{\phi_P}{\phi_P - \phi_W} & (Pe \rightarrow -\infty) \end{cases} \tag{C10}$$

In order to discriminate between the CDS, UDS and HS approaches to approximating the ES, we plot $\hat{Q}$ versus Pe. The plot depends, of course, on the values of ϕ_P and ϕ_W. To illustrate the point, and without loss of generality of the result, let us take $\phi_W = 0$ and $\phi_P = 1$. For this case the $\hat{Q}$ versus Pe relationships to plot are:

(i) CDS

$$\hat{Q} = \tfrac{1}{2}\, Pe - 1 \tag{C11}$$

(ii) UDS

$$\hat{Q} = \begin{cases} -1 & (u > 0) \\ Pe - 1 & (u < 0) \end{cases} \tag{C12}$$

(iii) HS

$$\hat{Q} = \begin{cases} \tfrac{1}{2}\, Pe - 1 & (|Pe| < 0) \\ 0 & (Pe \geq 2,\ u > 0) \\ Pe & (Pe \leq -2,\ u < 0) \end{cases} \tag{C13}$$

(iv) ES

$$\hat{Q} = -Pe \cdot \frac{e^{-Pe}}{1 - e^{-Pe}} \tag{C14}$$

The three limiting cases of the ES are also

$$\hat{Q} \rightarrow \begin{cases} -1 & (|Pe| \rightarrow 0) \\ 0 & (Pe \rightarrow +\infty) \\ -Pe & (Pe \rightarrow -\infty) \end{cases} \tag{C15}$$

In Fig. 13 are plotted four lines labelled CDS, UDS, HS and ES, each of which represents $\hat{Q}$ as a function of Pe. That marked ES represents the exact solution and its three limiting cases are clearly seen. When $|Pe|$ is small, ES tends to -1; this is the value appropriate to $\hat{Q}$ because of diffusion alone. When Pe is very large and positive, $\hat{Q}$ tends to 0; when Pe is very large in magnitude, but negative, $\hat{Q}$ tends to be equal to -Pe. With the CDS approach, $\hat{Q}$ tends to $\frac{1}{2}$Pe instead of zero when Pe is large and positive, and at the other extreme it tends to $\frac{1}{2}$Pe instead of Pe. These disagreements are quite severe. And, of course, the finite difference equation coefficients obtained with the CDS approach when $|Pe| > 2$ are such as to encourage divergence during an iterative solution procedure. The UDS approach of the other hand, has errors which are larger than the CDS approach for moderate values of Pe, but when $|Pe|$ is large the UDS solution is more exact than the CDS one.

The HS approach combines the advantages of the CDS and UDS ideas and its corresponding straight lines are marked on the figure. Evidently this prescription is somewhat more accurate, overall, than either of its associated predecessors. It clearly tends to the asymptotes of 0 and -Pe, as Pe tends to $+\infty$ and $-\infty$ respectively (making it superior to the UDS for large values of $|Pe|$), and is equivalent to the CDS approach where the latter may be usefully employed when $|Pe| < 2$. The HS approach also happens to keep the finite difference coefficients within bounds which ensure convergence of the iterative scheme for solving the resulting set of algebraic equations. Hence the recommendation and current use of the hybrid differencing scheme as given by Eq. (A13), and quoted earlier as Eq. (10).

APPENDIX D FORTRAN SYMBOL LIST

A(J)	=	Coefficient of recurrence relation
AE(I,J)	=	Coefficient of combined convective/diffusive flux through east-wall of control volume
AL1	=	X-coordinate of inlet boundary of flow domain
AL2	=	X-coordinate of outlet boundary of flow domain
ALAMDA	=	Length scale factor at inlet of flow domain
ALPHA	=	Inlet sloping wall expansion angle
ALTOT	=	Total length of pipe of larger diameter
AN(I,J)	=	Coefficient of combined convective/diffusive flux through north-wall of control volume
AP(I,J)	=	Sum of coefficients of combined convective/diffusive fluxes through all four walls of control volume
AREAEW	=	Area of east/west wall of control volume
AREAN	=	Area of north-wall of control volume
AREAS	=	Area of south-wall of control volume
ARDEN	=	Area of east/west cell-wall times density of fluid
ARDENT	=	Sum of all east/wall ARDEN at a cross-section
AS(I,J)	=	Coefficient of combined convective/diffusive flux through south-wall of control volume
AW(I,J)	=	Coefficient of combined convective/diffusive flux through west-wall of control volume
B(J)	=	Coefficient of recurrence formulae
C(J)	=	Coefficient of recurrence relation
C1	=	Constant of turbulence model (=1.44)
C2	=	Constant of turbulence model (=1.92)
CAPPA	=	Von Karman constant (=0.4187)

CD	=	Constant of turbulence model (=1.0)
CDTERM	=	CMU * * 0.25
CE	=	Coefficient of convective flux through east-wall of control volume
CMU	=	Constant of turbulence model (=0.09)
CN	=	Coefficient of convective flux through north-wall of control volume
CP	=	Maximum of zero and net outflow (SMP) from control volume
CPO	=	CP
CS	=	Coefficient of convective flux through south-wall of control volume
CW	=	Coefficient of convective flux through west-wall of control volume
D(J)	=	Coefficient of recurrence formulae
DE	=	Coefficient of diffusive flux through east-wall of control volume
DEN(I,J)	=	Density of fluid
DENSIT	=	Density of fluid at inlet of the calculation domain
DITERM	=	Coefficient of volume integral of energy dissipation rate in vicinity of walls
DN	=	Coefficient of diffusive flux through north-wall of control volume
DS	=	Coefficient of diffusive flux through south-wall of control volume
DU(I,K)	=	Coefficient of velocity-correction term for U velocity
DUDXE	=	$\partial u/\partial x$ at eastern face of U-cell
DUDXW	=	$\partial u/\partial x$ at western face of U-cell
DUDX	=	$\partial u/\partial x$ at main grid node (I,J)

DUDY	=	$\partial u/\partial y$ at main grid node (I,J)
DUDYE	=	$\partial u/\partial y$ at mid-point of east wall of V-cell
DUDYW	=	$\partial u/\partial y$ at mid-point of west wall of V-cell
DVDX	=	$\partial v/\partial x$ at main grid node (I,J)
DVDXN	=	$\partial v/\partial x$ at mid-point of north wall of U-cell
DVDXS	=	$\partial v/\partial x$ at mid-point of south wall of U-cell
DVDY	=	$\partial v/\partial y$ at main grid node (I,J)
DVDYN	=	$\partial v/\partial y$ at mid-point of north wall of V-cell
DVDYS	=	$\partial v/\partial y$ at mid-point of south wall of V-cell
DW	=	Coefficient of diffusive flux through west wall of control volume
DWDX	=	$\partial W/\partial x$ at main grid node (I,J)
DWDY	=	$\partial W/\partial y$ at main grid node (I,J)
DXEP(I)	=	X(I+1)-X(I)
DXEPU(I)	=	XU(I+1)-XU(I)
DXPW(I)	=	X(I)-X(I-1)
DYNP(J)	=	Y(J+1)-Y(J)
DYNPV(J)	=	YV(J+1)-YV(J)
DYPS(J)	=	Y(J)-Y(J-1)
DYPSV(J)	=	YV(J)-YV(J-1)
ED(I,J)	=	Energy dissipation rate, ε
EDIN	=	Energy dissipation rate at inlet of flow domain (ε_{in})
ELOG	=	Constant of P-function for heat transfer at walls (=9.793)
EPSX	=	Grid expansion factor in axial direction
FACTOR	=	Area ratio for setting initial u-velocity field
FLOW	=	Mass flow rate at a cross-section based on calculated velocity
FLOWIN	=	Total mass flow rate entering pipes

GE	=	Mass flux through east-wall of cell
GEN(I,J)	=	Generation of turbulence by shear from mean flow
GENCOU	=	Part of generation term modified in terms of wall shear stress
GENRES	=	Total unmodified generation of turbulence (GEN(I,J) less $\mu_t(\partial v/\partial x)^2$.
GN	=	Mass flux through north-wall of cell
GNW	=	Mass flux through north-wall of u-cell
GP	=	Mass flux at location of velocity
GREAT	=	A very large value (10^{30})
GS	=	Mass flux through south-wall of cell
GSW	=	Mass flux through south-wall of u-cell
HEDA	=	Heading 'KPLUS = TE * RHO/TAUN'
HEDB	=	Heading 'LENGTH SCALE/PIPE RADIUS'
HEDD	=	Heading 'ENERGY DISSIPATION'
HEDDK	=	Heading 'DIMENSIONLESS TURBULENCE ENERGY'
HEDDP	=	Heading 'DIMENSIONLESS PRESSURE'
HEDDSL	=	Heading 'DIMENSIONLESS STREAMLINE COORDS'
HEDDU	=	Heading 'DIMENSIONLESS U VELOCITY'
HEDDV	=	Heading 'DIMENSIONLESS V VELOCITY'
HEDDVS	=	Heading 'DIMENSIONLESS EFF. VISCOSITY'
HEDDW	=	Heading 'DIMENSIONLESS W VELOCITY'
HEDK	=	Heading 'TURBULENCE ENERGY'
HEDM	=	Heading 'VISCOSITY'
HEDP	=	Heading 'PRESSURE'
HEDSF	=	Heading 'DIMENSIONLESS STREAM FUNCTION'
HEDSL	=	Heading 'RADIAL COORDINATE OF STREAMLINES'
HEDT	=	Heading 'TEMPERATURE'

HEDU	=	Heading 'U VELOCITY'
HEDV	=	Heading 'V VELOCITY'
I	=	Index for dependent variables, and co-ordinates
IFINE	=	Logical parameter for fine x-direction grid spacing
IMON	=	I-index of monitoring location
INCALA	=	Additional (unused) logical parameter for selection of dependent variables
INCALB	=	Additional (unused) logical parameter for selection of dependent variables
INCALD	=	Logical parameter for solution of ε-equation
INCALK	=	Logical parameter for solution of k-equation
INCALM	=	Additional (unused) logical parameter for selection of dependent variables
INCALP	=	Logical parameter for solution of P'-equation
INCALS	=	Logical parameter for calculation of stream function
INCALU	=	Logical parameter for solution of U-equation
INCALV	=	Logical parameter for solution of V-equation
INCALW	=	Logical parameter for solution of W-equation
INDCOS	=	Control index for definition of co-ordinate system (= 1 for plane flows; = 2 for axisymmetric flows)
INITAL	=	Logical parameter to print initial field estimates
INPLOT	=	Logical parameter to produce line printer streamline plots
IREAD	=	Logical parameter to read initial field values from allocated disk storage
IWRITE	=	Logical parameter to write solution field values to allocated disk storage
INPRO	=	Logical parameter for updating of fluid properties
IPREF	=	I-index of location where pressure is fixed

IPRINT	=	If equal to NITER, activates printing of residual sums and monitor values of field variables
ISTEP	=	I-index of entrance plane, within calculation domain
ISTM1	=	ISTEP-1
ISTP1	=	ISTEP+1
IT	=	I-index of maximum dimension of dependent variables
J	=	Index for dependent variables, and co-ordinate
JMAX(I)	=	Maximum value of j-index within flow domain
JMON	=	J-index of monitoring location
JPREF	=	J-index of location where pressure is fixed
JPRINT	=	If equal to NITER, activates printing of field variable values
JSTEP	=	J-index of horizontal plane next to wall of, and within, smaller pipe
JSTM1	=	JSTEP-1
JSTP1	=	JSTEP+1
JT	=	J-index of maximum dimension of dependent variables
LFS	=	Index for counting loops for swirl
LFSMAX	=	Number of swirl loops to be run
MAXIT	=	Maximum number of iterations to be completed if iteration sequence is not stopped by test on value of SORCE
NI	=	Maximum value of I-index
NIM1	=	NI-1
NITER	=	Number of iterations completed
NJ	=	Maximum value of J-index
NJM1	=	NJ-1
NJM2	=	NJ-2

NONDIM	-	Activates the calculation and printing of dimensionless solution when specified as true
NSBR	=	Zero value specifies flat W profile; the value one specifies solid body rotation
NSTLN	=	Number of streamlines calculated
NSWPD	=	Number of application of line iteration for ε-equation
NSWPK	=	Number of application of line iteration for k-equation
NSWPP	=	Number of application of line iteration for P'-equation
NSWPU	=	Number of application of line iteration for U-equation
NSWPW	=	Number of application of line iteration for W-equation
P(I,J)	=	Pressure, P
PHI(I,J)	=	General representation for all dependent variables, ϕ
PP(I,J)	=	Pressure-correction, P'
PRANDT	=	Turbulent Prandtl number
PRED	=	Constant of turbulence model in ε-equation, σ_{ε}
PRTE	=	Constant of turbulence model in k-equation, σ_{ε}
PSTAR(I,J)	=	Dimensionless pressure
R(J)	=	Radius of main grid node (I,J) from symmetry axis
RCV(J)	=	Radius of C- and U-cell center
RESO R	=	Residual source for individual control volume
RESORE	=	Sum of residual sources within calculation domain for ε-equation
RSORK	=	Sum of residual sources within calculation domain for k-equation
RESORM	=	Sum of mass sources within calculation domain
RESORU	=	Sum of residual sources within calculation domain for U-equation
RESORV	=	Sum of residual sources within calculation domain for V-equation
RESORW	=	Sum of residual sources within calculation domain
RLARGE	=	Radius of large pipe

RSDRL	=	RSMALL/RLARGE
RSMALL	=	Radius of small pipe
RV(J)	=	Radius of location of V(I,J) from symmetry axis
SEW(I)	=	0.5*(DXEP(I) + DXPW(I))
SEWU(I)	=	0.5*(DXEPU(I) + DXPWU(I))
SMP	=	Net outflow from control volume
SNS(J)	=	0.5*(DYNP(J) + DYPS(J))
SNSV(J)	=	0.5*(DYNPV(J) + DYPSV(J))
SORCE	=	Maximum of RESORM, RESORU, RESORV, RESORW, RESORK
SORMAX	=	Maximum acceptable value of SORCE for converged solution
SORVOL	=	GREAT * VOL
SP(I,J)	=	Coefficient of linearized source treatment
SPKD(I,J)	=	-CP, for k- and ε-equations
SSC	=	Shear-stress coefficient
STFN(I,J)	=	Dimensionless stream function
STVAL(K)	=	Stream function value of streamlines
SU(I,J)	=	Coefficient of linearized source treatment
SUKD(I,J)	=	CPO * TE(I,J), for k-equation
	=	CPO * ED(I,J), for ε-equation
SWNB(LFS)	=	Inlet swirl number speciacation of WINST
SWRLNO	=	Calculated inlet swirl number
TAUN(I)	=	Shear stress at north wall-boundary of flow domain
TAURX	=	North wall shear stress, x-component
TAURW	=	North wall shear stress, θ-component
TAUW(J)	=	Shear stress at west wall-boundary of flow domain
TAUXR	=	West wall shear stress, r-component
TAUXW	=	West wall shear stress, θ-component
TE(I,J)	=	Turbulence energy, k

TESTAR(I,J)	=	Dimensionless turbulence energy
TEIN	=	Turbulence energy at inlet of flow domain (k_{in})
TMULT	=	Coefficient of wall shear-stress expression
TURBIN	=	Turbulence intensity factor at inlet of flow domain
U(I,J)	=	Component of mean velocity in axial direction (u-velocity)
UEFF	=	SQRT[U(I,J)**2 + W(I,J)**2]
UIN	=	U-velocity at inlet of flow domain
UINC	=	Uniform increment of u-velocity at outlet of flow domain
ULARGE	=	UIN * (RSMALL/RLARGE)**2
UMEAN	=	Mean u-velocity at inlet
URFE	=	Under-relaxation factor for energy dissipation
URFK	=	Under-relaxation factor for turbulence energy
URFP	=	Under-relaxation factor for pressure-correction
URFU	=	Under-relaxation factor for u-velocity
URFV	=	Under-relaxation factor for v-velocity
URFVIS	=	Under-relaxation factor for viscosity
URFW	=	Under-relaxation factor for w-velocity
USTAR(I,J)	=	Dimensionless u-velocity
V(I,J)	=	Component of mean velocity in radial direction (v-velocity)
VANB(LFS)	=	Swirl vane angle
VAVG	=	Average v-velocity between nodes (I,J) and (I,J+1)
VDR	=	V(I,J)/RV(J)
VIS(I,J)	=	Effective viscosity
VISCOS	=	Laminar viscosity
VISE	=	Effective viscosity at mid-point of east-wall of cell

VISOLD	=	Value of effective viscosity before underrelaxation
VISN	=	Effective viscosity at mid-point of north-wall of cell
VISS	=	Effective viscosity at mid-point of south-wall of cell
VISTAR(I,J)	=	Dimensionless effective viscosity
VISW	=	Effective viscosity at mid-point of west-wall of cell
VOL	=	Volume of cell or control-volume
VSTAR(I,J)	=	Dimensionless v-velocity
W(I,J)	=	w-velocity
WIN	=	Inlet w-velocity from swirl vanes
WINST	=	Inlet w-velocity at JSTEP from solid body rotation swirl generator
WMONIN	=	Inlet swirl momentum
WSTAR(I,J)	=	Dimensionless w-velocity
X(I)	=	Distance from inlet plane in axial direction
XMONIN	=	Momentum of fluid at inlet of flow domain
XND(I)	=	Dimensionless X(I)
XPLUSW(I)	=	Local Reynolds number based on friction velocity and distance from west wall-boundary of flow domain
XU(I)	=	X-coordinate of storage location of U(I,J)
XUDPLT(I)	=	Dimensionless XU(I) required for line printer streamline plots
Y(J)	=	Distance from symmetry axis in radial direction
YND(J)	=	Dimensionless Y(J)
YPLUSN(J)	=	Local Reynolds number based on friction velocity and distance from north wall-boundary of flow domain
YSLOPE	=	Height of first stairstep along sloping wall
YSLPLT(N,I)	=	Dimensionless YSTLN(I,K) required for line printer streamline plots

YSTLN(I,K)	=	Radial coordinate of points representing streamlines
YSTLND(I,K)	=	Dimensionless YSTLN(I,K)
YV(J)	=	Y-coordinate of storage location of V(I,J)
YUND(J)	=	Dimensionless YV(J)

APPENDIX E COMPUTER PROGRAM LISTING

The computer program listing obtained for the sample computation described in Section 4.3 is now given.

```
C      SUBROUTINE CONTRO
C
CA*********************************************************************
C
C
C      A COMPUTER PROGRAM FOR TURBULENT, SWIRLING, RECIRCULATING,
C      FLOW IN COMBUSTOR GEOMETRIES
C
C      VERSION OF APRIL, 1981
C
C      D L RHODE & D G LILLEY
C      MECHANICAL AND AEROSPACE ENGINEERING
C      OKLAHOMA STATE UNIVERSITY
C      STILLWATER, OK     74078
C
C
CA*********************************************************************
CHAPTER  0  0  0  0  0  0  0  0  PRELIMINARIES  0  0  0  0  0  0  0  0
C
       DIMENSION HEDU(9), HEDV(9), HEDW(9), HEDP(9), HEDT(9), HEDK(9),
      *HEDD(9), HEDM(9), HEDL(9), VANB(7),SWNB(7),HEDSF(9),HEDSL(9),
      #HEDDU(9),HEDDV(9),HEDDW(9),
      #HEDDP(9),HEDDK(9),HEDDSL(9),HEDDVS(9)
       DIMENSION YAXES(10),SYMB L(10)
       COMMON
      1/UVEL/RESORU,NSWPU,URFU,DXEPU(48),DXPWU(48),SEWU(48)
      1/VVEL/RESORV,NSWPV,URFV,DYNPV(24),DYPSV(24),SNSV(24)
      */WVEL/ RESORW, NSWPW, URFW
      1/PCOR/RESORM,NSWPP,URFP,DU(48,24),DV(48,24),IPREF,JPREF
      1/TEN/RESORK,NSWPK,URFK
      1/TDIS/RESORE,NSWPD,URFE
      */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),
      *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
      *VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
      #,VISTAR(48,24)
      1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
      1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
      1      SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
      #  WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
      #YND(24),YVND(24)
       COMMON
      1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
      1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
      2       RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
      1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE
      1/WALLF/YPLUSN(48),XPLUSW(24),TAUN(48),TAUW(24)
      1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),
      1      SP(48,24)
      1/PLOTT/NSTLN,NPLTLN,NPTS,YSLPLT(10,48),XUDPLT(48),INPLOT
       LOGICAL INCALU,INCALV,INCALW,INCALP,INPRO,INCALK,INCALD,INCALM,
      *INCALA,INCALB,INCALS,INPLOT,IWRITE,NONDIM,IREAD,INLET,
      *INITAL,IFINE
C---------ALL PRIMARY USER INPUTS ARE LOCATED HERE
       DATA VANB /0.,45.,55.,60.,65.,68.,70./
      #,SWNB/0.,.5,1.0,1.25,1.50,1.75,2.0/
       DATA  XAXIS /3HXI /
       DATA YAXES /3H 00,3H 02,3H 04,3H 06,3H 08,3H 10,
      #3H 06,3H 07,3H 08,3H 09/
       DATA SYMB L /1H0,1H2,1H4,1H6,1H8,1H1,1H6,1H7,1H8,1H9/
C---------DELETE UNDERFLOW ERROR MESSAGES, CALL TRAPS WHEN USING WATFIV
```

```
C---------CALL TRAPS(1,1,4000)
      CALL ERRSET(208,256,-1,1,0,0)
C---------SET INPLOT=.TRUE. ONLY FOR STREAMLINE LINE-PRINTER PLOT
      INPLOT=.TRUE.
C---------SET IWRITE=.TRUE. ONLY FOR WRITING SOLN. ON DATA FILES
      IWRITE=.FALSE.
C---------SET NONDIM=.TRUE. ONLY FOR PRINTING DIMENSIONLESS SOLN.
      NONDIM=.TRUE.
C---------SET IREAD=.TRUE. ONLY FOR READING INITIAL GUESS OF SOLN.
C---------     FROM DATA FILES
      IREAD=.FALSE.
C---------SET INITAL=.TRUE. ONLY FOR PRINTING INITIAL GUESS OF SOLN.
      INITAL=.TRUE.
C---------SET IFINE=.TRUE. ONLY FOR FINE GRID IN X-DIRECTION
      IFINE=.FALSE.
      IF(.NOT. IREAD) GO TO 150
      READ(12) X
      READ(12) Y
  150 CONTINUE
      NSTLN=11
      NPLTLN=6
      MAXLN=10
      NITER=0
      JPRINT=NITER+300
      IPRINT=NITER+1
      LFS=1
      LFSMAX=7
      NSBR=0
      MAXIT=NITER+200
C---------SEE STATEMENT 304 FOR MAXIT SPEC. FOR FURTHER SWIRL CASES
      DENSIT=1.211
      IT=48
      JT=24
      GREAT=1.E30
      NSWPU=4
      NSWPV=3
      NSWPW=3
      NSWPP=5
      NSWPK=3
      NSWPD=3
      READ(5,10) HEDU,HEDV,HEDW,HEDP,HEDT,HEDK,HEDD,HEDM,HEDL,
     $HEDSF,HEDSL,HEDDU,HEDDV,HEDDW,HEDDP,HEDDK,HEDDSL,HEDDVS
  010 FORMAT(9A4)
C
CHAPTER  1  1  1  1  1  PARAMETERS AND CONTROL INDICES  1  1  1  1  1  1
C
C---------GRID
      ISTEP=2
      JSTEP=8
      INDCOS=2
      NJ=21
      NJM1=NJ-1
      ISTP1=ISTEP+1
      ISTM1=ISTEP-1
      JSTP1=JSTEP+1
      JSTM1=JSTEP-1
      RLARGE=.0625
      ALTOT=.375
      IF(IFINE) GO TO 120
C---------COARSE MESH GRID LINES IN X-DIRECTION
```

```
      NI=20
      NIM1=NI-1
      EPSX=1.11
      IF(EPSX-1.)  13,12,13
13    SUMX=0.5*EPSX**(NI-4)+(EPSX**(NI-3)-1.)/(EPSX-1.)+0.5
      GO TO 15
12    CONTINUE
      SUMX=NIM1-1
15    DX=ALTOT/SUMX
      X(1)=-.5*DX
      X(2)=-X(1)
      DO  100  I=3,NIM1
      X(I)=X(I-1)+DX
  100 DX=EPSX*DX
      X(NI)=X(NIM1)+(X(NIM1)-X(NI-2))
      X(21)=X(20)+(X(20)-X(19))
      X(22)=X(21)+(X(20)-X(19))
      X(23)=X(22)+(X(20)-X(19))
      NI=23
      NIM1=NI-1
      ALTOT=(X(22)+X(23))/2.
      AL1=0.5*(X(ISTEP)+X(ISTM1))
      AL2=ALTOT-AL1
C---------FINE MESH GRID LINES IN X-DIRECTION
      IF(.NOT. IFINE) GO TO 130
  120 NI=30
      NIM1=NI-1
      EPSX=1.102
      IF(EPSX-1.)  17,16,17
17    SUMX=0.5*EPSX**(NI-4)+(EPSX**(NI-3)-1.)/(EPSX-1.)+0.5
      GO TO 18
16    CONTINUE
      SUMX=NIM1-1
18    DX=ALTOT/SUMX
      X(1)=-.5*DX
      X(2)=-X(1)
      DO  170  I=3,NIM1
      X(I)=X(I-1)+DX
  170 DX=EPSX*DX
      X(NI)=X(NIM1)+(X(NIM1)-X(NI-2))
      DO 180 L=31,35
  180 X(L)=X(L-1)+(X(NI)-X(NIM1))
      NI=35
      NIM1=NI-1
      ALTOT=(X(NIM1)+X(NI))/2.
      AL1=0.5*(X(ISTEP)+X(ISTM1))
      AL2=ALTOT-AL1
C---------SPECIFY RADIAL HEIGHT(NO. OF J-CELLS) OF COMBUSTOR
C---------          WALL FOR EACH I GRID LINE
  130 CONTINUE
      JMAX(1)=JSTEP
      JMAXP1(1)=JMAX(1)+1
      DO 160 I=2,NI
      JMAX(I)=JMAX(I-1)+3
      IF(JMAX(I-1) .EQ. NJM1) JMAX(I)=JMAX(I-1)
  160 JMAXP1(I)=JMAX(I)+1
C---------GRID LINES IN Y-DIRECTION
      Y(1)=-1.5625E-3
      Y(2)=1.5625E-3
      Y(3)=5.625E-3
```

```
      Y(4)=1.0625E-2
      Y(5)=1.625E-2
      Y(6)=2.1875E-2
      Y(7)=2.6875E-2
      Y(8)=3.03125E-2
      Y(9)=3.21875E-2
      Y(10)=3.4375E-2
      Y(11)=3.71875E-2
      Y(12)=4.09375E-2
      Y(13)=4.328125E-2
      Y(14)=4.5625E-2
      Y(15)=4.8125E-2
      Y(16)=5.078125E-2
      Y(17)=5.34375E-2
      Y(18)=5.59375E-2
      Y(19)=5.9375E-2
      Y(20)=6.1875E-2
      Y(21)=6.3125E-2
      RSMALL=0.5*(Y(JSTEP)+Y(JSTP1))
C---------DEPENDENT VARIABLE SELECTION
      INCALU=.TRUE.
      INCALV=.TRUE.
      INCALW=.TRUE.
      INCALP=.TRUE.
      INCALK=.TRUE.
      INCALD=.TRUE.
      INPRO=.TRUE.
      INCALS=.TRUE.
C---------FLUID PROPERTIES
C---------TURBULENCE CONSTANTS
      CMU=0.09
      CD=1.00
      C1=1.44
      C2=1.92
      CAPPA=.4187
      ELOG=9.793
      PRED=CAPPA*CAPPA/(C2-C1)/(CMU**.5)
      PRTE=1.0
C---------BOUNDARY VALUES
      UIN=30.
      ULARGE=UIN*(RSMALL/RLARGE)**2
      TURBIN=.03
      TEIN=TURBIN*UIN**2
      ALAMDA=0.005
      EDIN=TEIN**1.5/(ALAMDA*RLARGE)
      VISCOS=1.8E-5
C---------PRESSURE CALCULATION
      IPREF=2
      JPREF=2
C---------PROGRAM CONTROL AND MONITOR
      IMON=NIM1
      JMON=8
      SORMAX=.004
C
CHAPTER  2  2  2  2  2  2  INITIAL OPERATIONS  2  2  2  2  2  2  2  2  2
      CALL INIT
C---------NONDIMENSIONALIZE X & Y VARIABLES FOR NONDIMENSIONAL OUTPUT
      DO 50 I=1,NI
      XND(I)=X(I)/(2.*RLARGE)
   50 XUND(I)=XU(I)/(2.*RLARGE)
```

```
      DO 60 J=1,NJ
      YND(J)=Y(J)/(2.*RLARGE)
   60 YVND(J)=YV(J)/(2.*RLARGE)
C---------INITIALISE VARIABLE FIELDS
      FLOWIN=0.0
      ARDEN=0.0
      ARDENT=0.
      XMONIN=0.
      WMONIN=0.0
      ANGMOM=0.
C
C---------INLET SWIRL VELOCITY PROFILE
C
C---------W,  USE SOLID BODY ROTATION MODEL
      WINST=2.*SWNB(LFS)/(1.+SWNB(LFS))*UIN
      DO 206 J=2,JSTEP
  206 W(1,J)=WINST*R(J)/R(JSTEP)
C---------NSBR=0  -  FLAT SWIRL VELOCITY PROFILE FROM SWIRL VANES
C---------NSBR=1  -  SOLID BODY ROTATION FROM SWIRL GENERATOR
      IF(NSBR .EQ. 1) GO TO 208
C---------W, FLAT PROFILE
      WIN=UIN*TAN(VANB(LFS)*3.14159/180.)
      DO 207 J=2,JSTEP
  207 W(1,J)=WIN
  208 CONTINUE
C---------INITIALIZE U-,TE-,ED-, & W-FIELDS
      DO  200  J=2,JSTEP
      U(2,J)=UIN
      TE(1,J)=TEIN
      ED(1,J)=EDIN
      ARDEN=0.5*(DEN(1,J)+DEN(2,J))*RCV(J)*SNS(J)
      XMONIN=XMONIN+ARDEN*U(2,J)*U(2,J)
      WMONIN=WMONIN+ARDEN*U(2,J)*W(1,J)
      ANGMOM=ANGMOM+ARDEN*U(2,J)*W(1,J)*R(J)
      ARDENT=ARDENT+ARDEN
  200 FLOWIN=FLOWIN+ARDEN*U(2,J)
      UMEAN=FLOWIN/ARDENT
      SWRLNO=ANGMOM/(XMONIN*RSMALL)
      IF(W(1,JSTEP) .EQ. 0.) WMONIN=1.
      DO  202  I=2,NI
      IJ=JMAXP1(I-1)
      FACTOR=(YV(JSTP1)*RV(JSTP1))/(YV(IJ)*RV(IJ))
      JJ=JMAX(I-1)
      DO  202  J=2,JJ
      U(I,J)=UIN*FACTOR
202   CONTINUE
C
      IF(NSBR .EQ. 0) GO TO 219
      DO 209 I=2,NI
      JJ=JMAX(I)
      DO 209 J=2,JJ
      WINST=2.*SWNB(LFS)/(1.+SWNB(LFS))*U(I,2)
      W(I,J)=WINST*R(J)/R(JJ)
      TE(I,J)=TEIN
      ED(I,J)=EDIN
  209 CONTINUE
      GO TO 221
C
219   CONTINUE
      DO 220 I=2,NI
```

```
      JJ=JMAX(I)
      DO 220 J=2,JJ
      TE(I,J)=TEIN
      ED(I,J)=EDIN
  220 CONTINUE
C
  221 CONTINUE
      DO  203  I=2,NIM1
  203 YPLUSN(I)=11.0
      DO  204  J=JSTEP,NJ
      XPLUSW(J)=11.0
  204 IF(J.EQ.JSTEP) XPLUSW(J)= 0.0
      URFVIS=.7
      CALL PROPS
C---------INITIAL OUTPUT
      WRITE(6,211)
      IK=JMAXP1(ISTEP)
      YSLOPE=YV(IK)-YV(JSTP1)
      IF(JMAX(ISTEP) .LT. NJM1) ALPHA=ATAN(YSLOPE/SEWU(2))*180./3.14159
      IF((JSTEP .LT. NJM1) .AND. (JMAX(ISTEP) .EQ. NJM1)) ALPHA=90.
      WRITE(6,225) ALPHA
      WRITE(6,235) RSMALL
      WRITE(6,240) RLARGE
      WRITE(6,245) ALTOT
      RE=UIN*RSMALL*2.0*DENSIT/VISCOS
      WRITE(6,250) RE
      WRITE(6,255) VISCOS
      RSDRL=RSMALL/RLARGE
      WRITE(6,260) DENSIT
  295 CONTINUE
      IF(.NOT. IWRITE) GO TO 297
      WRITE(11) X
      WRITE(11) Y
      WRITE(14) XUND
  297 CONTINUE
      IF(.NOT. IREAD) GO TO 298
      READ(12) U
      READ(12) V
      READ(12) W
      READ(12) P
      READ(12) TE
      READ(12) ED
      READ(12) VIS
      READ(12) STFN
  298 CONTINUE
      IF(.NOT. INITAL) GO TO 299
      IF(INCALU) CALL PRINT(1,1,NI,NJ,IT,JT,XU,Y,U,HEDU)
      IF(INCALV) CALL PRINT(1,1,NI,NJ,IT,JT,X,YV,V,HEDV)
      IF(INCALW) CALL PRINT(1, 1, NI, NJ, IT, JT, X, Y, W, HEDW)
      IF(INCALK) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,TE,HEDK)
      IF(INCALD) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,ED,HEDD)
  299 CONTINUE
      RESORU=.005
      URFP=1.
      RESORV=.005
      RESORW=.005
C
CHAPTER  3  3  3  3  3  3  3  ITERATION LOOP  3  3  3  3  3  3  3  3  3
C
      WRITE(6,310) IMON,JMON
```

```
  300 NITER=NITER+1
      DO 330 I=2,NIM1
      DO 330 J=2,NJM1
  330 PP(I,J)=0.
      IF(LFS .GE. 3) GO TO 425
      URFU=.5
      URFV=.5
      URFW=.6
      URFK=.7
      URFE=.7
      URFVIS=.7
      IF(LFS .LT. 3) GO TO 430
  425 CONTINUE
C---------INCREASE UNDERRELAXATION FACTORS AS CONVERGENCE NEARS
      URFU=.15+(FLOAT(NITER))*((.35-.15)/40.)
      IF(URFU .GT. .35) URFU=.35
      URFV=.20+(FLOAT(NITER))*((.25-.20)/40.)
      IF(URFV .GT. .25) URFV=.25
      URFW=.50+(FLOAT(NITER))*((.60-.50)/40.)
      IF(URFW .GT. .60) URFW=.60
      URFK=.70
      URFE=.70
      URFVIS=.70
      IF((RESORV .LT. .08) .AND. (RESORU .LT. .10)) URFU=.40
      IF((RESORV .LT. .06) .AND. (RESORU .LT. .08)) URFU=.45
      IF((RESORK .LT. .10E-1) .AND. (RESORE .LT. .10E+10)) URFE=.75
      IF((RESORK .LT. .4E-2) .AND. (RESORE .LT. .5E+9)) URFE=.80
      IF(RESORU .LT. SORMAX) URFU=.20
      IF(RESORV .LT. SORMAX) URFV=.20
      IF(RESORW .LT. SORMAX) URFW=.25
      IF(RESORK .LT. .10E-2) URFK=.60
      IF(RESORE .LT. .10E+8) URFE=.55
  430 CONTINUE
C---------UPDATE MAIN DEPENDENT VARIABLES
      IF(INCALU) CALL CALCU
      IF(INCALV) CALL CALCV
      IF(INCALP) CALL CALCP
      IF(INCALW) CALL CALCW
      IF(INCALK) CALL CALCTE
      IF(INCALD) CALL CALCED
C---------UPDATE FLUID PROPERITIES
      IF(INPRO) CALL PROPS
C---------INTERMEDIATE OUTPUT
      RESORM=RESORM/FLOWIN
      RESORU=RESORU/XMONIN
      RESORV=RESORV/XMONIN
      RESORW=RESORW/WMONIN
      RESORK=RESORK/(.5*FLOWIN*UMEAN*UMEAN)
      IF(NITER .NE. IPRINT) GO TO 301
      IPRINT=IPRINT+1
      WRITE(6,311) NITER,RESORU,RESORV,RESORW,RESORM,RESORK,
     *RESORE,U(IMON,JMON),V(IMON,JMON),W(IMON,JMON),P(IMON,NJM1),
     *ED(IMON,NJM1)
      IF(NITER .NE. JPRINT) GO TO 301
      IF(INCALU) CALL PRINT(1,1,NI,NJ,IT,JT,XU,Y,U,HEDU)
      IF(INCALV) CALL PRINT(1,1,NI,NJ,IT,JT,X,YV,V,HEDV)
      IF(INCALW) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,W,HEDW)
      IF(INCALP) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,P,HEDP)
      IF(INCALD) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,ED,HEDD)
      JPRINT=JPRINT+25
```

```
      WRITE(6,310) IMON,JMON
  301 CONTINUE
C---------TERMINATION TESTS
      SORCE=AMAX1(RESORM,RESORU,RESORV,RESORW,RESORK)
      IF(NITER.GE.MAXIT) GO TO 302
  303 IF(NITER .GE. 150 .AND. SORCE .GE. 3.0) GO TO 302
      IF(SORCE.GT.SORMAX .OR. NITER .LT. 20) GO TO 300
  302 CONTINUE
      IF(NITER .GE. 150 .AND. SORCE .GE. 3.0) WRITE(6,960)
C
CHAPTER  4  4  4  4  4  4  FINAL OPERATIONS AND OUTPUT  4  4  4  4  4  4
C
  440 CONTINUE
C---------NONDIMENSIONALIZE PROBLEM SOLN.
      IF(.NOT. NONDIM) GO TO 700
      DO 600 I=1,NI
      DO 600 J=1,NJ
      USTAR(I,J)=U(I,J)/UIN
      VSTAR(I,J)=V(I,J)/UIN
      WSTAR(I,J)=W(I,J)/UIN
      PSTAR(I,J)=P(I,J)/(DENSIT*(UIN**2)/2.)
      TESTAR(I,J)=TE(I,J)/(UIN*UIN)
      SP(I,J)=0.0
      IF(ED(I,J) .GT. 1.E-15) SP(I,J)=TE(I,J)**1.5/ED(I,J)/RLARGE
      VISTAR(I,J)=VIS(I,J)/VISCOS
  600 CONTINUE
  700 CONTINUE
      IF(INCALS) CALL STRMFN
      WRITE(6,312)
      WRITE(6,410) LFS,NSBR,SWNB(LFS),VANB(LFS),SWRLNO,UMEAN,FLOWIN
      IF(INCALU) CALL PRINT(1,1,NI,NJ,IT,JT,XU,Y,U,HEDU)
      IF(INCALV) CALL PRINT(1,1,NI,NJ,IT,JT,X,YV,V,HEDV)
      IF(INCALW) CALL PRINT(1, 1, NI, NJ, IT, JT, X, Y, W, HEDW)
      IF(INCALP) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,P,HEDP)
      IF(INCALP) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,PP,HEDP)
      IF(INCALK) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,TE,HEDK)
      IF(INCALD) CALL PRINT(1,1,NI,NJ,IT,JT,X,Y,ED,HEDD)
      IF(INCALS) CALL PRINT(1,1,NI,NJ,IT,JT,XU,Y,STFN,HEDSF)
      IF(INCALS) CALL PRINT(1,1,NI,NSTLN,IT,JT,XU,STVAL,YSTLN,HEDSL)
      IF(.NOT. NONDIM) GO TO 750
      IF(INCALU) CALL PRINT(1,1,NI,NJ,IT,JT,XUND,YND,USTAR,HEDDU)
      IF(INCALV) CALL PRINT(1,1,NI,NJ,IT,JT,XND,YVND,VSTAR,HEDDV)
      IF(INCALW) CALL PRINT(1, 1, NI, NJ, IT, JT, XND, YND,WSTAR,HEDDW)
      IF(INCALS) CALL PRINT(1,1,NI,NSTLN,IT,JT,XUND,STVAL,YSTLND,HEDDSL)
      IF(INCALP) CALL PRINT(1,1,NI,NJ,IT,JT,XND,YND,PSTAR,HEDDP)
      IF(INCALK) CALL PRINT(1,1,NI,NJ,IT,JT,XND,YND,TESTAR,HEDDK)
      IF(INCALK) CALL PRINT(2,2,NI,NJ,IT,JT,X,Y,SP,HEDL)
      IF(INCALK) CALL PRINT(1,1,NI,NJ,IT,JT,XND,YND,VISTAR,HEDDVS)
  750 CONTINUE
      IF(.NOT. IWRITE) GO TO 702
      WRITE(11) U
      WRITE(11) V
      WRITE(11) W
      WRITE(11) P
      WRITE(11) TE
      WRITE(11) ED
      WRITE(11) VIS
      WRITE(11) STFN
      WRITE(14) YSTLND
  702 CONTINUE
```

```
C---------CALCULATION OF SHEAR-STRESS COEFFICIENT ALONG LARGE DUCT WALL
      WRITE(6,402)
      DO  401  I=2,NIM1
      SSC=ABS(TAUN(I))/(.5*DENSIT*UIN*UIN)
      WRITE(6,403) I,XND(I),SSC
  401 CONTINUE
      WRITE(6,312)
C---------PLOT DIMENSIONLESS STREAMLINES
      LARGE=0
      IF(INPLOT .AND. INCALS) CALL PLOT (XUDPLT,IT,NPTS,XAXIS,YSLPLT,
     #MAXLN,NPLTLN,YAXES,SYMB L,LARGE)
      LARGE=1
      IF(INPLOT .AND. INCALS) CALL PLOT (XUDPLT,IT,NPTS,XAXIS,YSLPLT,
     #MAXLN,NPLTLN,YAXES,SYMB L,LARGE)
C---------RESET INITIAL CONDITIONS FOR ANOTHER SWIRL CASE
      IF(LFS .GE. LFSMAX) GO TO 409
      LFS=LFS+1
      NITER=0
      JPRINT=NITER+300
      IPRINT=NITER+1
  304 IF(LFS .GE. 3) MAXIT=NITER+200
      IF(NSBR .EQ. 0) GO TO 405
      WINST=2.*SWNB(LFS)/(1.+SWNB(LFS))*UIN
      DO 406 J=2,JSTEP
  406 W(1,J)=WINST*R(J)/R(JSTEP)
      GO TO 408
  405 WIN=UIN*TAN(VANB(LFS)*3.14159/180.)
      DO 407 J=2,JSTEP
  407 W(1,J)=WIN
  408 FLOWIN=0.
      ARDEN=0.
      ARDENT=0.
      XMONIN=0.
      ANGMOM=0.
      WMONIN=0.
C---------READ INITIAL GUESS OF NEXT SWIRL PROBLEM FROM
C---------PREVIOUS SOLN. OF SIMILAR PROBLEM
      IF(.NOT. IREAD) GO TO 445
      READ(12) U
      READ(12) V
      READ(12) W
      READ(12) P
      READ(12) TE
      READ(12) ED
      READ(12) VIS
      READ(12) STFN
  445 CONTINUE
      DO  490  J=2,JSTEP
      ARDEN=0.5*(DEN(1,J)+DEN(2,J))*RCV(J)*SNS(J)
      XMONIN=XMONIN+ARDEN*U(2,J)*U(2,J)
      WMONIN=WMONIN+ARDEN*U(2,J)*W(1,J)
      ANGMOM=ANGMOM+ARDEN*U(2,J)*W(1,J)*R(J)
      ARDENT=ARDENT+ARDEN
  490 FLOWIN=FLOWIN+ARDEN*U(2,J)
      UMEAN=FLOWIN/ARDENT
      SWRLNO=ANGMOM/(XMONIN*RSMALL)
      IF(W(1,JSTEP) .EQ. 0.) WMONIN=1.
      WRITE(6,310) IMON,JMON
      GO TO 300
  409 CONTINUE
```

```
      STOP
C---------FORMAT STATEMENTS
  211 FORMAT(1H1,T37,'AXISYMMETRIC,ISOTHERMAL, GT COMBUSTOR FLOWFIELD SI
     #MULATION',//,T35,'USING THE STAIRSTEP APPROXIMATION FOR THE SLOPIN
     #G EXPANSION WALL',//,T53,'AND THE K-E TURBULENCE MODEL')
  225 FORMAT(////,T40,'EXPANSION ANGLE(DEG.) =',T77,1PE13.3)
  230 FORMAT(//,T40,'NUMBER OF STAIRSTEPS =',T81,I1)
  235 FORMAT (//,T40,'INLET RADIUS(M) =',T77,1PE13.3)
  240 FORMAT(//,T40,'COMBUSTOR RADIUS(M) =',T77,1PE13.3)
  245 FORMAT(//,T40,'COMBUSTOR LENGTH(M) =',T77,1PE13.3)
  250 FORMAT(//,T40,'INLET REYNOLDS NO.(USING DIAM.) =',T77,1PE13.3)
  255 FORMAT(//,T40,'LAMINAR VISCOSITY(KG/M/SEC) =',T77,1PE13.3)
  260 FORMAT(//,T40,'DENSITY(KG/CU. M) =',T77,1PE13.3,////)
  310 FORMAT(13H0ITER    I---,  9X,29HABSOLUTE RESIDUAL SOURCE SUMS,9X,
     111H---I   I---,37H FIELD VALUES AT MONITORING LOCATION(,I2,1H,,I2,
     *6H) ---I/14H  NO      UMON,7X,'VMON',7X,'WMON',7X,'MASS',7X,
     *'TKIN'
     3,7X,4HDISP,9X,1HU,9X,1HV,10X,1HW,10X,1HP,10X,1HD/)
  311 FORMAT (2X,I4,11E11.4)
  312 FORMAT (1H0,59(2H- ))
  402 FORMAT(///9X,1HI,5X,3HX/D,5X,10HS.S.COEFF.)
  403 FORMAT(/5X,I5,2(1PE11.3))
  410 FORMAT(//23H SWIRL CASE WITH LFS  =,I3/
     1          23H                AND NSBR =,I3//
     163H CORRESPONDS IF NSBR = 1 TO SWIRL GENERATOR WITH SWIRL NUMBER =
     1,F10.3//37H OR IF NSBR = 0 TO SWIRL VANE ANGLE =,F10.3//,
     11X,' COMPUTED INLET SWIRL NUMBER =',F10.4//,
     11X,' COMPUTED INLET MEAN AXIAL VELOCITY =',F10.4//,
     11X,' COMPUTED INLET MASS FLOW RATE =',F10.5////)
  950 FORMAT(10X,6F10.5,/,10X,6F10.5,/,10X,4F10.5)
  960 FORMAT(15X,'  THE SOLN. IS NOT CONVERGING')
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE INIT
CA*********************************************************************
C
CHAPTER  0  0  0  0  0  0  0  0  PRELIMINARIES  0  0  0  0  0  0  0  0
C
      COMMON
     1/UVEL/RESORU,NSWPU,URFU,DXEPU(48),DXPWU(48),SEWU(48)
     1/VVEL/RESORV,NSWPV,URFV,DYNPV(24),DYPSV(24),SNSV(24)
     */WVEL/ RESORW, NSWPW, URFW
     1/PCOR/RESORM,NSWPP,URFP,DU(48,24),DV(48,24),IPREF,JPREF
     */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),
     *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
     *VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
     #,VISTAR(48,24)
     1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
     1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
     1       SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
     #  WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
     #YND(24),YVND(24)
      COMMON
     1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
     1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
     2        RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
     1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE
     1/WALLF/YPLUSN(48),XPLUSW(24),TAUN(48),TAUW(24)
```

```
     1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),
     1      SP(48,24)
     1/PLOTT/NSTLN,NPLTLN,NPTS,YSLPLT(10,48),XUDPLT(48),INPLOT
C
CHAPTER  1  1  1  1  1  CALCULATE GEOMETRICAL QUANTITIES  1  1  1  1  1
C
      DO 100 J=1,NJ
      R(J)=Y(J)
  100 IF(INDCOS.EQ.1)R(J)=1.0
      DXPW(1)=0.0
      DXEP(NI)=0.0
      DO 101 I=1,NIM1
      DXEP(I)=X(I+1)-X(I)
  101 DXPW(I+1)=DXEP(I)
      DYPS(1)=0.0
      DYNP(NJ)=0.0
      DO 102 J=1,NJM1
      DYNP(J)=Y(J+1)-Y(J)
  102 DYPS(J+1)=DYNP(J)
      SEW(1)=0.0
      SEW(NI)=0.0
      DO 103 I=2,NIM1
  103 SEW(I)=0.5*(DXEP(I)+DXPW(I))
      SNS(1)=0.0
      SNS(NJ)=0.0
      DO 104 J=2,NJM1
  104 SNS(J)=0.5*(DYNP(J)+DYPS(J))
      XU(1)=0.0
      DO 105 I=2,NI
  105 XU(I)=0.5*(X(I)+X(I-1))
      DXPWU(1)=0.0
      DXPWU(2)=0.0
      DXEPU(1)=0.0
      DXEPU(NI)=0.0
      DO 106 I=2,NIM1
      DXEPU(I)=XU(I+1)-XU(I)
  106 DXPWU(I+1)=DXEPU(I)
      SEWU(1)=0.0
      DO 107 I=2,NI
  107 SEWU(I)=X(I)-X(I-1)
C---------U-VELOCITIES WEIGHTING FACTORS
      DO 111  I=2,NIM1
      WFE(I)=SEWU(I+1)/(SEWU(I+1)+SEWU(I))
      IF(I .LE. 2) GO TO 111
      WFW(I)=SEWU(I-1)/(SEWU(I-1)+SEWU(I))
  111 CONTINUE
      YV(1)=0.0
      RV(1)=0.0
      DO 108 J=2,NJ
      RV(J)=0.5*(R(J)+R(J-1))
  108 YV(J)=0.5*(Y(J)+Y(J-1))
      RCV(1)=R(1)
      RCV(NJ)=R(NJ)
      DO 113 J=2,NJM1
  113 RCV(J)=0.5*(RV(J+1)+RV(J))
      DYPSV(1)=0.0
      DYPSV(2)=0.0
      DYNPV(NJ)=0.0
      DO 109 J=2,NJM1
      DYNPV(J)=YV(J+1)-YV(J)
```

```
  109 DYPSV(J+1)=DYNPV(J)
      SNSV(1)=0.0
      DO 110 J=2,NJ
  110 SNSV(J)=Y(J)-Y(J-1)
C---------V-VELOCITIES WEIGHTING FACTORS
      DO 112 J=3,NJM1
      WFN(J)=SNSV(J+1)/(SNSV(J+1)+SNSV(J))
      WFS(J)=SNSV(J-1)/(SNSV(J-1)+SNSV(J))
  112 CONTINUE
C
CHAPTER  2  2  2  2  2  2  SET VARIABLES TO ZERO  2  2  2  2  2  2
C
      DO 200 I=1,NI
      TAUN(I)=1.0
      DO 200 J=1,NJ
      TAUW(J)=1.0
      U(I,J)=0.0
      V(I,J)=0.0
      W(I,J)=0.
      P(I,J)=0.0
      PP(I,J)=0.0
      TE(I,J)=0.0
      ED(I,J)=0.0
      DEN(I,J)=DENSIT
      VIS(I,J)=VISCOS
      DU(I,J)=0.0
      DV(I,J)=0.0
      SU(I,J)=0.0
      SP(I,J)=0.0
      STFN(I,J)=0.
200   CONTINUE
      DO 300 I=1,NI
      DO 300 J=1,NSTLN
      YSTLN(I,J)=0.
      YSTLND(I,J)=0.
      STVAL(J)=0.
  300 CONTINUE
      DO 400 N=1,NPLTLN
      DO 400 I=1,NI
      YSLPLT(N,I)=0.
  400 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE PROPS
CA*******************************************************************
C
CHAPTER  0  0  0  0  0  0  0  0  PRELIMINARIES  0  0  0  0  0  0  0  0
C
      COMMON
     1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
     */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),
     *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
     *VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
     #,VISTAR(48,24)
     1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
     1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE
C
```

```
CHAPTER   1   1   1   VISCOSITY   1   1   1
C
      DO 100 I=2,NIM1
      DO 100 J=2,NJM1
      VISOLD=VIS(I,J)
      IF(ED(I,J).EQ.0.) GO TO 102
      VIS(I,J)=DEN(I,J)*TE(I,J)**2*CMU/ED(I,J)+VISCOS
      GO  TO  101
  102 VIS(I,J)=VISCOS
C---------UNDER-RELAX VISCOSITY
101   VIS(I,J)=URFVIS*VIS(I,J)+(1.-URFVIS)*VISOLD
  100 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE CALCU
CA*****************************************************************
C
CHAPTER  0  0  0  0  0  0  0  0  PRELIMINARIES  0  0  0  0  0  0  0  0
C
      COMMON
     1/UVEL/RESORU,NSWPU,URFU,DXEPU(48),DXPWU(48),SEWU(48)
     1/VVEL/RESORV,NSWPV,URFV,DYNPV(24),DYPSV(24),SNSV(24)
     1/PCOR/RESORM,NSWPP,URFP,DU(48,24),DV(48,24),IPREF,JPREF
     */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),
     *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
     *VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
     #,VISTAR(48,24)
     1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
     1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
     1      SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
     #  WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
     #YND(24),YVND(24)
      COMMON
     1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
     1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),
     1      SP(48,24)
     1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
     2       RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
C
CHAPTER  1  1  1  1  1  1  ASSEMBLY OF COEFFICIENTS  1  1  1  1  1  1  1
C
      DO 100 I=3,NIM1
      DO 101 J=2,NJM1
C---------COMPUTE AREAS AND VOLUME
      AREAN=RV(J+1)*SEWU(I)
      AREAS=RV(J)*SEWU(I)
      AREAEW=RCV(J)*SNS(J)
      VOL=RCV(J)*SEWU(I)*SNS(J)
C---------CALCULATE CONVECTION COEFFICIENTS
      GN=0.5*(DEN(I,J+1)+DEN(I,J))*V(I,J+1)
      GNW=0.5*(DEN(I-1,J)+DEN(I-1,J+1))*V(I-1,J+1)
      GS=0.5*(DEN(I,J-1)+DEN(I,J))*V(I,J)
      GSW=0.5*(DEN(I-1,J)+DEN(I-1,J-1))*V(I-1,J)
      GE=DEN(I,J)*(U(I+1,J)*(1.0-WFE(I))+U(I,J)*WFE(I))
      GW=DEN(I-1,J)*(U(I-1,J)*(1.0-WFW(I))+U(I,J)*WFW(I))
      CN=0.5*(GN+GNW)*AREAN
      CS=0.5*(GS+GSW)*AREAS
```

```
      CE=GE*AREAEW
      CW=GW*AREAEW
C---------CALCULATE DIFFUSION COEFFICIENTS
      VISN=0.25*(VIS(I,J)+VIS(I,J+1)+VIS(I-1,J)+VIS(I-1,J+1))
      VISS=0.25*(VIS(I,J)+VIS(I,J-1)+VIS(I-1,J)+VIS(I-1,J-1))
      DN=VISN*AREAN/DYNP(J)
      DS=VISS*AREAS/DYPS(J)
      DE=VIS(I,J)*AREAEW/DXEPU(I)
      DW=VIS(I-1,J)*AREAEW/DXPWU(I)
C---------CALCULATE COEFFICIENTS OF SOURCE TERMS
      SMP=CN-CS+CE-CW
      CP=AMAX1(0.0,SMP)
      CPO=CP
C---------ASSEMBLE MAIN COEFFICIENTS
      AN(I,J)=AMAX1(ABS(0.5*CN),DN)-0.5*CN
      AS(I,J)=AMAX1(ABS(0.5*CS),DS)+0.5*CS
      DE=AMAX1(DE,-WFE(I)*CE,(1.0-WFE(I))*CE)
      DW=AMAX1(DW,WFW(I)*CW,-(1.0-WFW(I))*CW)
      AE(I,J)=DE-(1.0-WFE(I))*CE
      AW(I,J)=DW+(1.-WFW(I))*CW
      DU(I,J)=AREAEW
      DUDXE=(U(I+1,J)-U(I,J))/DXEPU(I)
      DUDXW=(U(I,J)-U(I-1,J))/DXPWU(I)
      SORCE1=(DUDXE*VIS(I,J)-DUDXW*VIS(I-1,J))/SEWU(I)
      DVDXN=(V(I,J+1)-V(I-1,J+1))/SEWU(I)
      DVDXS=(V(I,J)-V(I-1,J))/SEWU(I)
      SORCE2=(RV(J+1)*VISN*DVDXN-RV(J)*VISS*DVDXS)/(RCV(J)*DYNPV(J))
      SU(I,J)=CPO*U(I,J)+DU(I,J)*(P(I-1,J)-P(I,J))
      SU(I,J)=SU(I,J)+(SORCE1+SORCE2)*VOL
      SP(I,J)=-CP
  101 CONTINUE
  100 CONTINUE
C
CHAPTER  2  2  2  2  2  2  2  PROBLEM MODIFICATIONS  2  2  2  2  2  2  2
C
      CALL PROMOD (2)
C
CHAPTER  3  FINAL COEFF. ASSEMBLY AND RESIDUAL SOURCE CALCULATION  3  3
C
      RESORU=0.0
      DO 300 I=3,NIM1
      DO 301 J=2,NJM1
      AP(I,J)=AN(I,J)+AS(I,J)+AE(I,J)+AW(I,J)-SP(I,J)
      DU(I,J)=DU(I,J)/AP(I,J)
      RESOR=AN(I,J)*U(I,J+1)+AS(I,J)*U(I,J-1)+AE(I,J)*U(I+1,J)
     1     +AW(I,J)*U(I-1,J)-AP(I,J)*U(I,J)+SU(I,J)
      VOL=RCV(J)*SEWU(I)*SNS(J)
      SORVOL=GREAT*VOL
      IF(-SP(I,J).GT.0.5*SORVOL)  RESOR=RESOR/SORVOL
      RESORU=RESORU+ABS(RESOR)
C---------UNDER-RELAXATION
      AP(I,J)=AP(I,J)/URFU
      SU(I,J)=SU(I,J)+(1.-URFU)*AP(I,J)*U(I,J)
      DU(I,J)=DU(I,J)*URFU
  301 CONTINUE
  300 CONTINUE
C
CHAPTER  4  4  4  SOLUTION OF DIFFERENCE EQUATION  4  4  4  4  4  4  4
C
      DO 400 N=1,NSWPU
```

```
  400 CALL LISOLV(3,2,NI,JMAX,IT,JT,U,2)
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE CALCV
CA*******************************************************************
C
CHAPTER  0  0  0  0  0  0  0  0  PRELIMINARIES  0  0  0  0  0  0  0  0
C
      COMMON
     1/UVEL/RESORU,NSWPU,URFU,DXEPU(48),DXPWU(48),SEWU(48)
     1/VVEL/RESORV,NSWPV,URFV,DYNPV(24),DYPSV(24),SNSV(24)
     1/PCOR/RESORM,NSWPP,URFP,DU(48,24),DV(48,24),IPREF,JPREF
     */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),
     *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
     *VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
     #,VISTAR(48,24)
     1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
     1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
     1       SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
     #  WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
     #YND(24),YVND(24)
      COMMON
     1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
     1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),
     1       SP(48,24)
     1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
     2        RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
C
CHAPTER  1  1  1  1  1  1  ASSEMBLY OF COEFFICIENTS  1  1  1  1  1  1  1
C
      DO 100 I=2,NIM1
      DO 101 J=3,NJM1
C---------COMPUTE AREAS AND VOLUME
      AREAN=R(J)*SEW(I)
      AREAS=R(J-1)*SEW(I)
      AREAEW=RV(J)*SNSV(J)
      VOL=RV(J)*SEW(I)*SNSV(J)
C---------CALCULATE CONVECTION COEFFICIENTS
      GN=DEN(I,J)*(V(I,J+1)*(1.0-WFN(J))+V(I,J)*WFN(J))
      GS=DEN(I,J-1)*(V(I,J-1)*(1.0-WFS(J))+V(I,J)*WFS(J))
      GE=0.5*(DEN(I+1,J)+DEN(I,J))*U(I+1,J)
      GSE=0.5*(DEN(I,J-1)+DEN(I+1,J-1))*U(I+1,J-1)
      GW=0.5*(DEN(I,J)+DEN(I-1,J))*U(I,J)
      GSW=0.5*(DEN(I,J-1)+DEN(I-1,J-1))*U(I,J-1)
      CN=GN*AREAN
      CS=GS*AREAS
      CE=0.5*(GE+GSE)*AREAEW
      CW=0.5*(GW+GSW)*AREAEW
C---------CALCULATE DIFFUSION COEFFICIENTS
      VISE=0.25*(VIS(I,J)+VIS(I+1,J)+VIS(I,J-1)+VIS(I+1,J-1))
      VISW=0.25*(VIS(I,J)+VIS(I-1,J)+VIS(I,J-1)+VIS(I-1,J-1))
      DN=VIS(I,J)*AREAN/DYNPV(J)
      DS=VIS(I,J-1)*AREAS/DYPSV(J)
      DE=VISE*AREAEW/DXEP(I)
      DW=VISW*AREAEW/DXPW(I)
C---------CALCULATE COEFFICIENTS OF SOURCE TERMS
      SMP=CN-CS+CE-CW
```

```
      CP=AMAX1(0.0,SMP)
      CPO=CP
C---------ASSEMBLE MAIN COEFFICIENTS
      DN=AMAX1(DN,-WFN(J)*CN,(1.0-WFN(J))*CN)
      DS=AMAX1(DS,WFS(J)*CS,-(1.0-WFS(J))*CS)
      AN(I,J)=DN-(1.0-WFN(J))*CN
      AS(I,J)=DS+(1.0-WFS(J))*CS
      AE(I,J)=AMAX1(ABS(0.5*CE),DE)-0.5*CE
      AW(I,J)=AMAX1(ABS(0.5*CW),DW)+0.5*CW
      DV(I,J)=0.5*(AREAN+AREAS)
      DUDYE=(U(I+1,J)-U(I+1,J-1))/SNSV(J)
      DUDYW=(U(I,J)-U(I,J-1))/SNSV(J)
      SORCE1=(DUDYE*VISE-DUDYW*VISW)/DXEPU(I)
      DVDYN=(V(I,J+1)-V(I,J))/DYNPV(J)
      DVDYS=(V(I,J)-V(I,J-1))/DYPSV(J)
      SORCE2=(VIS(I,J)*RCV(J)*DVDYN-VIS(I,J-1)*RCV(J-1)*DVDYS)
     #  /(RV(J)*SNSV(J))
      SU(I,J)=CPO*V(I,J)+DV(I,J)*(P(I,J-1)-P(I,J))
      SU(I,J)=SU(I,J)+(SORCE1+SORCE2)*VOL
      SP(I,J)=-CP
      IF(INDCOS .EQ. 1) GO TO 101
      SU(I,J)=SU(I,J)+VOL*((DEN(I,J)+DEN(I,J-1))*(W(I,J)+W(I,J-1)
     #)**2)/(8.*RV(J))
      SP(I,J)=SP(I,J)-(VIS(I,J)+VIS(I,J-1))*VOL/RV(J)**2
  101 CONTINUE
  100 CONTINUE
C
CHAPTER  2  2  2  2  2  2  2  PROBLEM MODIFICATIONS  2  2  2  2  2  2  2
C
      CALL PROMOD (3)
C
CHAPTER  3  FINAL COEFF. ASSEMBLY AND RESIDUAL SOURCE CALCULATION  3  3
C
      RESORV=0.0
      DO 300 I=2,NIM1
      DO 301 J=3,NJM1
      AP(I,J)=AN(I,J)+AS(I,J)+AE(I,J)+AW(I,J)-SP(I,J)
      DV(I,J)=DV(I,J)/AP(I,J)
      RESOR=AN(I,J)*V(I,J+1)+AS(I,J)*V(I,J-1)+AE(I,J)*V(I+1,J)
     1      +AW(I,J)*V(I-1,J)-AP(I,J)*V(I,J)+SU(I,J)
      VOL=RV(J)*SEW(I)*SNSV(J)
      SORVOL=GREAT*VOL
      IF(-SP(I,J).GT.0.5*SORVOL)  RESOR=RESOR/SORVOL
      RESORV=RESORV+ABS(RESOR)
C---------UNDER-RELAXATION
      AP(I,J)=AP(I,J)/URFV
      SU(I,J)=SU(I,J)+(1.-URFV)*AP(I,J)*V(I,J)
      DV(I,J)=DV(I,J)*URFV
  301 CONTINUE
  300 CONTINUE
C
CHAPTER  4  4  4  SOLUTION OF DIFFERENCE EQUATION  4  4  4  4  4  4  4
C
      DO 400 N=1,NSWPV
  400 CALL LISOLV(2,3,NI,JMAX,IT,JT,V,3)
      RETURN
      END
C
C-----------------------------------------------------------------
C
```

```
      SUBROUTINE CALCW
CA*****************************************************************
C
CHAPTER  0  0  0  0  0  0  0  0  PRELIMINARIES  0  0  0  0  0  0  0  0
C
      COMMON
     1/UVEL/RESORU,NSWPU,URFU,DXEPU(48),DXPWU(48),SEWU(48)
     1/VVEL/RESORV,NSWPV,URFV,DYNPV(24),DYPSV(24),SNSV(24)
     */WVEL/ RESORW, NSWPW, URFW
     1/TEN/RESORK,NSWPK,URFK
     1/TDIS/RESORE,NSWPD,URFE
     */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),
     *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
     *VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
     #,VISTAR(48,24)
     1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
     1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
     1      SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
     #  WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
     #YND(24),YVND(24)
      COMMON
     1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
     1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
     2       RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
     1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE
     1/WALLF/YPLUSN(48),XPLUSW(24),TAUN(48),TAUW(24)
     1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),
     1      SP(48,24)
C---------IF NO SWIRL, RETURN TO MAIN
      IF(W(1,3) .LE. 0.) GO TO 500
C
CHAPTER 1 1 1 1 1 1 1 1 ASSEMBLY OF COEFFICIENTS 1 1 1 1 1 1 1
C
      DO 100  I=2, NIM1
      DO 101  J=2, NJM1
C
C       COMPUTE AREAS AND VOLUME
C
      AREAN=RV(J+1)*SEW(I)
      AREAS=RV(J)*SEW(I)
      AREAEW=RCV(J)*SNS(J)
      VOL=RCV(J)*SNS(J)*SEW(I)
C
C---------CALCULATE CONVECTION COEFFICIENTS
C
      GN=0.5*(DEN(I,J)+DEN(I,J+1))*V(I,J+1)
      GS=0.5*(DEN(I,J)+DEN(I,J-1))*V(I,J)
      GE=0.5*(DEN(I,J)+DEN(I+1,J))*U(I+1,J)
      GW=0.5*(DEN(I,J)+DEN(I-1,J))*U(I,J)
      CN=GN*AREAN
      CS=GS*AREAS
      CE=GE*AREAEW
      CW=GW*AREAEW
C
C---------CALCULATE DIFFUSION COEFFICIENTS
C
      VISN=0.5*(VIS(I,J)+VIS(I,J+1))
      VISS=0.5*(VIS(I,J)+VIS(I,J-1))
      VISE=0.5*(VIS(I,J)+VIS(I+1,J))
      VISW=0.5*(VIS(I,J)+VIS(I-1,J))
```

```
      DN=VISN*AREAN/DYNP(J)
      DS=VISS*AREAS/DYPS(J)
      DE=VISE*AREAEW/DXEP(I)
      DW=VISW*AREAEW/DXPW(I)
C
C---------SOURCE TERMS
C
      SMP=CN-CS+CE-CW
      CP=AMAX1(0., SMP)
      CPO=CP
C
C---------ASSEMBLE MAIN COEFFICIENTS
C
      AN(I,J)=AMAX1(ABS(0.5*CN),DN)-0.5*CN
      AS(I,J)=AMAX1(ABS(0.5*CS),DS)+0.5*CS
      AE(I,J)=AMAX1(ABS(0.5*CE),DE)-0.5*CE
      AW(I,J)=AMAX1(ABS(0.5*CW),DW)+0.5*CW
      DV=0.5*(AREAN+AREAS)
      VAVG=0.5* (V(I,J+1)+V(I,J))
      SU(I,J)=CPO*W(I,J)
      IF(INDCOS .EQ. 1) GO TO 101
      SORCE1=-DEN(I,J)*VAVG*W(I,J) /RCV(J)
      SORCE2=-(VISN*RV(J+1)-VISS*RV(J)) *W(I,J)/(DYNPV(J)*
     #RCV(J)*RCV(J))
      SU(I,J)=SU(I,J)+(SORCE1+SORCE2)*VOL
      SP(I,J)=-CP
  101 CONTINUE
100   CONTINUE
C
CHAPTER 2 2 2 2 2 2 PROBLEM MODIFICATIONS 2 2 2 2 2 2 2 2 2 2 2
C
      CALL PROMOD (8)
C
CHAPTER 3 3 FINAL COEFFICIENT ASSEMBLY AND RESIDUAL SOURCE CALCULATION
C
      RESORW=0.
      DO 300  I=2, NIM1
      DO 301  J=2,NJM1
      AP(I,J)=AN(I,J)+AS(I,J)+AE(I,J)+AW(I,J)-SP(I,J)
      RESOR=AN(I,J)*W(I,J+1)+AS(I,J)*W(I,J-1)+AE(I,J)*W(I+1,J)
     *  +AW(I,J)*W(I-1,J)-AP(I,J)*W(I,J)+SU(I,J)
      VOL=RCV(J)*SNS(J)*SEW(I)
      SORVOL=GREAT*VOL
      IF(-SP(I,J) .GT. 0.5*SORVOL) RESOR=RESOR/SORVOL
      IF(J .LE. 2) RESOR=0.
      RESORW=RESORW+ABS(RESOR)
C
C---------UNDER RELAXATION
C
      AP(I,J)=AP(I,J)/URFW
      SU(I,J)=SU(I,J)+(1.-URFW)*AP(I,J)*W(I,J)
301   CONTINUE
300   CONTINUE
C
CHAPTER 4 4 4 SOLUTION OF DIFFERENCE EQUATIONS 4 4 4 4 4 4 4 4
C
      DO 400  N=1, NSWPW
400   CALL LISOLV(2, 2, NI, JMAX, IT, JT, W,8)
  500 CONTINUE
      RETURN
```

```
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE CALCP
CA*********************************************************************
C
CHAPTER  0  0  0  0  0  0  0  0  PRELIMINARIES  0  0  0  0  0  0  0  0
C
      COMMON
     1/PCOR/RESORM,NSWPP,URFP,DU(48,24),DV(48,24),IPREF,JPREF
     */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),
     *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
     *VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
     #,VISTAR(48,24)
     1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
     1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
     1       SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
     #  WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
     #YND(24),YVND(24)
      COMMON
     1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
     1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),
     1       SP(48,24)
     1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
     2        RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
      RESORM=0.0
C
CHAPTER  1  1  1  1  1  1  ASSEMBLY OF COEFFICIENTS  1  1  1  1  1  1  1
C
      DO 100 I=2,NIM1
      DO 101 J=2,NJM1
C---------COMPUTE AREAS AND VOLUME
      AREAN=RV(J+1)*SEW(I)
      AREAS=RV(J)*SEW(I)
      AREAEW=RCV(J)*SNS(J)
      VOL=RCV(J)*SNS(J)*SEW(I)
C---------CALCULATE COEFFICIENTS
      DENN=0.5*(DEN(I,J)+DEN(I,J+1))
      DENS=0.5*(DEN(I,J)+DEN(I,J-1))
      DENE=0.5*(DEN(I,J)+DEN(I+1,J))
      DENW=0.5*(DEN(I,J)+DEN(I-1,J))
      AN(I,J)=DENN*AREAN*DV(I,J+1)
      AS(I,J)=DENS*AREAS*DV(I,J)
      AE(I,J)=DENE*AREAEW*DU(I+1,J)
      AW(I,J)=DENW*AREAEW*DU(I,J)
C---------CALCULATE SOURCE TERMS
      CN=DENN*V(I,J+1)*AREAN
      CS=DENS*V(I,J)*AREAS
      CE=DENE*U(I+1,J)*AREAEW
      CW=DENW*U(I,J)*AREAEW
      SMP=CN-CS+CE-CW
      SP(I,J)=0.0
      SU(I,J)=-SMP
C---------COMPUTE SUM OF ABSOLUTE MASS SOURCES
      RESORM=RESORM+ABS(SMP)
  101 CONTINUE
  100 CONTINUE
C
CHAPTER  2  2  2  2  2  2  2  PROBLEM MODIFICATIONS  2  2  2  2  2  2  2
```

```
C
      CALL PROMOD (4)
C
CHAPTER  3  3  3  3  3  FINAL COEFFICIENT ASSEMBLY  3  3  3  3  3  3  3
C
      DO 300 I=2,NIM1
      DO 301 J=2,NJM1
  301 AP(I,J)=AN(I,J)+AS(I,J)+AE(I,J)+AW(I,J)-SP(I,J)
  300 CONTINUE
C
CHAPTER  4  4  4  4  4  SOLUTION OF DIFFERENCE EQUATIONS  4  4  4  4  4
C
      DO 400 N=1,NSWPP
  400 CALL LISOLV(2,2,NI,JMAX,IT,JT,PP,4)
C
CHAPTER  5  5  5  5  CORRECT VELOCITIES AND PRESSURE  5  5  5  5  5  5
C
C---------VELOCITIES
      DO 503 I=2,NIM1
      JJ=JMAX(I)
      DO 501 J=3,JJ
      V(I,J)=V(I,J)+DV(I,J)*(PP(I,J-1)-PP(I,J))
  501 CONTINUE
      JJ=JMAX(I-1)
      DO 502 J=2,JJ
      IF(I .NE. 2) U(I,J)=U(I,J)+DU(I,J)*(PP(I-1,J)-PP(I,J))
  502 CONTINUE
  503 CONTINUE
C---------PRESSURES (WITH PROVISION FOR UNDER-RELAXATION)
      PPREF=PP(IPREF,JPREF)
      DO 506 I=2,NIM1
      JJ=JMAX(I)
      DO 508 J=2,JJ
      P(I,J)=P(I,J)+URFP*(PP(I,J)-PPREF)
C---------PP IS ZEROED AT TOP OF CHAPTER 3, MAIN
  508 CONTINUE
  506 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE CALCTE
CA*********************************************************************
C
CHAPTER  0  0  0  0  0  0  0  0  PRELIMINARIES  0  0  0  0  0  0  0  0
C
      COMMON
     1/TEN/RESORK,NSWPK,URFK
     */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),
     *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
     *VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
     #,VISTAR(48,24)
     1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
     1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
     1      SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
     #  WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
     #YND(24),YVND(24)
      COMMON
     1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
```

```
     1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),
     1      SP(48,24)
     1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE
     1/WALLF/YPLUSN(48),XPLUSW(24),TAUN(48),TAUW(24)
     1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
     2       RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
     1/SUSP/SUKD(48,24),SPKD(48,24)
C
CHAPTER  1  1  1  1  1  1  ASSEMBLY OF COEFFICIENTS  1  1  1  1  1  1
C
      PRTE=1.0
      DO 100 I=2,NIM1
      DO 101 J=2,NJM1
C---------COMPUTE AREAS AND VOLUME
      AREAN=RV(J+1)*SEW(I)
      AREAS=RV(J)*SEW(I)
      AREAEW=RCV(J)*SNS(J)
      VOL=RCV(J)*SNS(J)*SEW(I)
C---------CALCULATE CONVECTION COEFFICIENTS
      GN=0.5*(DEN(I,J)+DEN(I,J+1))*V(I,J+1)
      GS=0.5*(DEN(I,J)+DEN(I,J-1))*V(I,J)
      GE=0.5*(DEN(I,J)+DEN(I+1,J))*U(I+1,J)
      GW=0.5*(DEN(I,J)+DEN(I-1,J))*U(I,J)
      CN=GN*AREAN
      CS=GS*AREAS
      CE=GE*AREAEW
      CW=GW*AREAEW
C---------CALCULATE DIFFUSION COEFFICIENTS
      GAMN=0.5*(VIS(I,J)+VIS(I,J+1))/PRTE
      GAMS=0.5*(VIS(I,J)+VIS(I,J-1))/PRTE
      GAME=0.5*(VIS(I,J)+VIS(I+1,J))/PRTE
      GAMW=0.5*(VIS(I,J)+VIS(I-1,J))/PRTE
      DN=GAMN*AREAN/DYNP(J)
      DS=GAMS*AREAS/DYPS(J)
      DE=GAME*AREAEW/DXEP(I)
       DW=GAMW*AREAEW/DXPW(I)
C---------SOURCE TERMS
      SMP=CN-CS+CE-CW
      CP=AMAX1(0.0,SMP)
      CPO=CP
      DUDX=(U(I+1,J)-U(I,J))/SEW(I)
      DVDY=(V(I,J+1)-V(I,J))/SNS(J)
      DUDY=((U(I,J)+U(I+1,J)+U(I,J+1)+U(I+1,J+1))/4.-(U(I,J)+U(I+1,J)+
     1U(I,J-1)+U(I+1,J-1))/4.)/SNS(J)
      DVDX=((V(I,J)+V(I,J+1)+V(I+1,J)+V(I+1,J+1))/4.-(V(I,J)+V(I,J+1)+V(
     1I-1,J)+V(I-1,J+1))/4.)/SEW(I)
      DWDY=(W(I,J+1)-W(I,J-1))/(DYNP(J)+DYPS(J))-W(I,J)/R(J)
      DWDX=(W(I+1,J)-W(I-1,J))/(DXPW(I)+DXEP(I))
      GEN(I,J)=(2.*(DUDX**2+DVDY**2)+(DUDY+DVDX)**2)*VIS(I,J)
      IF(INDCOS .EQ. 2) GEN(I,J)=GEN(I,J)+VIS(I,J)*(DWDY**2+DWDX**2)
      IF(RV(J) .EQ. 0.) GO TO 110
      VDR=V(I,J)/RV(J)
      IF(INDCOS .EQ. 2) GEN(I,J)=GEN(I,J)+VIS(I,J)*.5*
     #(VDR+V(I,J+1)/RV(J+1))**2
      GO TO 120
  110 IF(INDCOS .EQ. 2) GEN(I,J)=GEN(I,J)+VIS(I,J)*
     #0.5*(V(I,J+1)/RV(J+1))**2
  120 CONTINUE
C---------ASSEMBLE MAIN COEFFICIENTS
      AN(I,J)=AMAX1(ABS(0.5*CN),DN)-0.5*CN
```

```
      AS(I,J)=AMAX1(ABS(0.5*CS),DS)+0.5*CS
      AE(I,J)=AMAX1(ABS(0.5*CE),DE)-0.5*CE
      AW(I,J)=AMAX1(ABS(0.5*CW),DW)+0.5*CW
      SU(I,J)=CPO*TE(I,J)
      SUKD(I,J)=SU(I,J)
      SU(I,J)=SU(I,J)+GEN(I,J)*VOL
      SP(I,J)=-CP
      SPKD(I,J)=SP(I,J)
      SP(I,J)=SP(I,J)-CD*CMU*DEN(I,J)**2*TE(I,J)*VOL/VIS(I,J)
  101 CONTINUE
  100 CONTINUE
C
CHAPTER  2  2  2  2  2  2  PROBLEM MODIFICATIONS  2  2  2  2  2  2
C
      CALL PROMOD (6)
C
CHAPTER 3 FINAL COEFFICIENT ASSEMBLY AND RESIDUAL SOURCE CALCULATION 3
C
      RESORK=0.0
      DO 300 I=2,NIM1
      DO 301 J=2,NJM1
      AP(I,J)=AN(I,J)+AS(I,J)+AE(I,J)+AW(I,J)-SP(I,J)
      RESOR=AN(I,J)*TE(I,J+1)+AS(I,J)*TE(I,J-1)+AE(I,J)*TE(I+1,J)
     1      +AW(I,J)*TE(I-1,J)-AP(I,J)*TE(I,J)+SU(I,J)
      VOL=RCV(J)*SNS(J)*SEW(I)
      SORVOL=GREAT*VOL
      IF(-SP(I,J).GT.0.5*SORVOL)  RESOR=RESOR/SORVOL
      RESORK=RESORK+ABS(RESOR)
C---------UNDER-RELAXATION
      AP(I,J)=AP(I,J)/URFK
      SU(I,J)=SU(I,J)+(1.-URFK)*AP(I,J)*TE(I,J)
  301 CONTINUE
  300 CONTINUE
C
CHAPTER  4  4  4  4  4  SOLUTION OF DIFFERENCE EQUATIONS  4  4  4  4  4
C
      DO 400 N=1,NSWPK
  400 CALL LISOLV(2,2,NI,JMAX,IT,JT,TE,6)
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE CALCED
CA*********************************************************************
C
CHAPTER  0  0  0  0  0  0  0  0  PRELIMINARIES  0  0  0  0  0  0  0  0
C
      COMMON
     1/TDIS/RESORE,NSWPD,URFE
     1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
     1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
     1      SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
     #  WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
     #YND(24),YVND(24)
     1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
     1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),
     1      SP(48,24)
     */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),
     *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
```

```
     *VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
     #,VISTAR(48,24)
      COMMON
     1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE
     1/WALLF/YPLUSN(48),XPLUSW(24),TAUN(48),TAUW(24)
     1/SUSP/SUKD(48,24),SPKD(48,24)
     1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
     2       RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
C
CHAPTER  1  1  1  1  1  1  ASSEMBLY OF COEFFICIENTS  1  1  1  1  1  1
C
      DO 100 I=2,NIM1
      JJ=JMAX(I)
      DO 101 J=2,JJ
C---------COMPUTE AREAS AND VOLUME
      AREAN=RV(J+1)*SEW(I)
      AREAS=RV(J)*SEW(I)
      AREAEW=RCV(J)*SNS(J)
      VOL=RCV(J)*SNS(J)*SEW(I)
C---------CALCULATE CONVECTION COEFFICIENTS
      GN=0.5*(DEN(I,J)+DEN(I,J+1))*V(I,J+1)
      GS=0.5*(DEN(I,J)+DEN(I,J-1))*V(I,J)
      GE=0.5*(DEN(I,J)+DEN(I+1,J))*U(I+1,J)
      GW=0.5*(DEN(I,J)+DEN(I-1,J))*U(I,J)
      CN=GN*AREAN
      CS=GS*AREAS
      CE=GE*AREAEW
      CW=GW*AREAEW
C---------CALCULATE DIFFUSION COEFFICIENTS
      GAMN=0.5*(VIS(I,J)+VIS(I,J-1))/PRED
      GAMS=0.5*(VIS(I,J)+VIS(I,J-1))/PRED
      GAME=0.5*(VIS(I,J)+VIS(I+1,J))/PRED
      GAMW=0.5*(VIS(I,J)+VIS(I-1,J))/PRED
      DN=GAMN*AREAN/DYNP(J)
      DS=GAMS*AREAS/DYPS(J)
      DE=GAME*AREAEW/DXEP(I)
       DW=GAMW*AREAEW/DXPW(I)
C---------SOURCE TERMS
      SMP=CN-CS+CE-CW
      CP=AMAX1(0.0,SMP)
      CPO=CP
C---------ASSEMBLE MAIN COEFFICIENTS
      AN(I,J)=AMAX1(ABS(0.5*CN),DN)-0.5*CN
      AS(I,J)=AMAX1(ABS(0.5*CS),DS)+0.5*CS
      AE(I,J)=AMAX1(ABS(0.5*CE),DE)-0.5*CE
      AW(I,J)=AMAX1(ABS(0.5*CW),DW)+0.5*CW
      SU(I,J)=CPO*ED(I,J)
      SUKD(I,J)=SU(I,J)
      SU(I,J)=SU(I,J)+C1*CMU*GEN(I,J)*VOL*DEN(I,J)*TE(I,J)/VIS(I,J)
      SP(I,J)=-CP
      SPKD(I,J)=SP(I,J)
      SP(I,J)=SP(I,J)-C2*DEN(I,J)*ED(I,J)*VOL/TE(I,J)
  101 CONTINUE
  100 CONTINUE
C
CHAPTER  2  2  2  2  2  2  PROBLEM MODIFICATIONS  2  2  2  2  2  2
C
      CALL PROMOD (7)
C
CHAPTER 3 FINAL COEFFICIENT ASSEMBLY AND RESIDUAL SOURCE CALCULATION 3
```

```
C
      RESORE=0.0
      DO 300 I=2,NIM1
      DO 301 J=2,NJM1
      AP(I,J)=AN(I,J)+AS(I,J)+AE(I,J)+AW(I,J)-SP(I,J)
      RESOR=AN(I,J)*ED(I,J+1)+AS(I,J)*ED(I,J-1)+AE(I,J)*ED(I+1,J)
     1      +AW(I,J)*ED(I-1,J)-AP(I,J)*ED(I,J)+SU(I,J)
      VOL=RCV(J)*SNS(J)*SEW(I)
      SORVOL=GREAT*VOL
      IF(-SP(I,J).GT.0.5*SORVOL)  RESOR=RESOR/SORVOL
      RESORE=RESORE+ABS(RESOR)
C---------UNDER-RELAXATION
      AP(I,J)=AP(I,J)/URFE
      SU(I,J)=SU(I,J)+(1.-URFE)*AP(I,J)*ED(I,J)
  301 CONTINUE
  300 CONTINUE
C
CHAPTER  4  4  4  4  4  SOLUTION OF DIFFERENCE EQUATIONS  4  4  4  4  4
C
      DO 400 N=1,NSWPD
  400 CALL LISOLV(2,2,NI,JMAX,IT,JT,ED,7)
      RETURN
      END
C
C-------------------------------------------------------------------
C
      SUBROUTINE LISOLV(ISTART,JSTART,NI,JMAX,IT,JT,PHI,NCHAP)
CA**********************************************************************
C
CHAPTER  0  0  0  0  0  0  0  0  PRELIMINARIES  0  0  0  0  0  0  0  0
C
      DIMENSION PHI(IT,JT),A(48),B(48),C(48),D(48),JMAX(IT)
      COMMON
     1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),
     1      SP(48,24)
     1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
     2       RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
     JSM1=JSTART-1
     NIM1=NI-1
     A(JSM1)=0.0
C---------COMMENCE W-E SWEEP
      DO 100 I=ISTART,NIM1
      C(JSM1)=PHI(I,JSM1)
C---------COMMENCE S-N TRAVERSE
      JI=JMAX(I)
      IF(NCHAP .EQ. 2) JI=JMAX(I-1)
      DO 101 J=JSTART,JI
C---------ASSEMBLE TDMA COEFFICIENTS
      A(J)=AN(I,J)
      B(J)=AS(I,J)
      C(J)=AE(I,J)*PHI(I+1,J)+AW(I,J)*PHI(I-1,J)+SU(I,J)
      D(J)=AP(I,J)
C---------CALCULATE COEFFICIENTS OF RECURRENCE FORMULA
      TERM=1./(D(J)-B(J)*A(J-1))
      A(J)=A(J)*TERM
      C(J)=(C(J)+B(J)*C(J-1))*TERM
  101 CONTINUE
C---------OBTAIN NEW PHI)S
      DO 102 JJ=JSTART,JI
      J=JI+1+JSM1-JJ
```

```
  102 PHI(I,J)=A(J)*PHI(I,J+1)+C(J)
  100 CONTINUE
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE PRINT(ISTART,JSTART,NI,NJ,IT,JT,X,Y,PHI,HEAD)
CA*******************************************************************
C
      DIMENSION PHI(IT,JT),X(IT),Y(JT),HEAD(9),STORE(48)
      ISKIP=1
      JSKIP=1
      WRITE(6,110)HEAD
      ISTA=ISTART-12
  100 CONTINUE
      ISTA=ISTA+12
      IEND=ISTA+11
      IF(NI.LT.IEND)IEND=NI
      WRITE(6,111)(I,I=ISTA,IEND,ISKIP)
      WRITE(6,114)(X(I),I=ISTA,IEND,ISKIP)
      WRITE(6,112)
      DO 101 JJ=JSTART,NJ,JSKIP
      J=JSTART+NJ-JJ
      DO 120 I=ISTA,IEND
      A=PHI(I,J)
      IF(ABS(A).LT.1.E-20) A=0.0
120   STORE(I)=A
  101 WRITE(6,113)J,Y(J),(STORE(I),I=ISTA,IEND,ISKIP)
      IF(IEND.LT.NI)GO TO 100
      RETURN
  110 FORMAT(1H0,17(2H*-),7X,9A4,7X,17(2H-*))
  111 FORMAT(1H0,13H       I =    ,I2,11I9)
  112 FORMAT(8H0 J     Y)
  113 FORMAT(I3,0PF8.5,1X,1P12E9.2)
  114 FORMAT(11H        X = ,F8.5,11F9.5)
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE PROMOD (NCHAP)
CA*******************************************************************
C
CHAPTER  0  0  0  0  0  0  0  0  PRELIMINARIES  0  0  0  0  0  0  0  0
C
      COMMON
     1/UVEL/RESORU,NSWPU,URFU,DXEPU(48),DXPWU(48),SEWU(48)
     1/VVEL/RESORV,NSWPV,URFV,DYNPV(24),DYPSV(24),SNSV(24)
     */WVEL/ RESORW, NSWPW, URFW
     */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),
     *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
     *VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
     #,VISTAR(48,24)
     1/PCOR/RESORM,NSWPP,URFP,DU(48,24),DV(48,24),IPREF,JPREF
     1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
     1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
     1       SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
     #  WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
     #YND(24),YVND(24)
      COMMON
```

```
     1/FLUPR/URFVIS,VISCOS,DENSIT,PRANDT,DEN(48,24),VIS(48,24)
     1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
     2        RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
     1/SUSP/SUKD(48,24),SPKD(48,24)
     1/COEF/AP(48,24),AN(48,24),AS(48,24),AE(48,24),AW(48,24),SU(48,24),
     1       SP(48,24)
     1/TURB/GEN(48,24),CD,CMU,C1,C2,CAPPA,ELOG,PRED,PRTE
     1/WALLF/YPLUSN(48),XPLUSW(24),TAUN(48),TAUW(24)
      IF(NCHAP .EQ. 2) GO TO 1150
      IF(JSTEP .EQ. NJM1) GO TO 1150
C---------OUT OF RANGE VALUES
      DO 1100 I=2,NI
      IF(JMAX(I) .EQ. NJM1) GO TO 1150
      JJ=JMAXP1(I)
      DO 1100 J=JJ,NJM1
 1100 SP(I,J)=-GREAT
 1150 CONTINUE
      GO TO (1,2,3,4,5,6,7,8),NCHAP
C
C
C
CHAPTER  1  1  1  1  1  1  1  1  PROPERTIES  1  1  1  1  1  1  1  1  1
C
    1 CONTINUE
C---------NO MODIFICATIONS FOR THIS PROBLEM
      RETURN
C
CHAPTER  2  2  2  2  2  2  2  2  U MOMENTUM  2  2  2  2  2  2  2  2  2
C
    2 CONTINUE
C---------OUT OF RANGE VALUES
      IF(JSTEP .EQ. NJM1) GO TO 202
      DO 200 I=3,NI
      IF(JMAX(I-1) .EQ. NJM1) GO TO 202
      JJ=JMAXP1(I-1)
      DO 200 J=JJ,NJM1
      SP(I,J)=-GREAT
  200 CONTINUE
  202 CONTINUE
C---------TOP WALL
      CDTERM=CMU**0.25
      DO 210 I=3,NIM1
      J=JMAX(I-1)
      YP=YV(J+1)-Y(J)
      SQRTK=SQRT(0.5*(TE(I,J)+TE(I-1,J)))
      DENU=0.5*(DEN(I,J)+DEN(I-1,J))
      YPLUSA=0.5*(YPLUSN(I)+YPLUSN(I-1))
      IF(YPLUSA.LE.11.63)     GO   TO   211
      TMULT=DENU*CDTERM*SQRTK*CAPPA/ALOG(ELOG*YPLUSA)
      GO   TO   212
  211 TMULT=VISCOS/YP
  212 CONTINUE
  205 SP(I,J)=SP(I,J)-TMULT*SEWU(I)*RV(J+1)
      IF(JMAX(I-1) .NE. JMAX(I)) SP(I,J)=SP(I,J)/2.
  210 AN(I,J)=0.
C---------SIDE WALL
      IF(JSTEP .EQ. NJM1) GO TO 214
      DO 225 I=3,NIM1
      IF(JMAX(I-2) .GE. JMAX(I-1)) GO TO 225
      JJ=JMAXP1(I-2)
```

```
      JI=JMAX(I-1)
      DO 220 J=JJ,JI
      AW(I,J)=0.
  220 CONTINUE
  225 CONTINUE
C---------SYMMETRY  AXIS
  214 CONTINUE
      DO 203 I=1,NI
  203 AS(I,2)=0.
C---------OUTLET
      ARDENT=0.0
      FLOW=0.0
      DO 209 J=2,NJM1
      ARDEN=0.50*(DEN(NIM1,J)+DEN(NIM1-1,J))*RCV(J)*SNS(J)
      ARDENT=ARDENT+ARDEN
  209 FLOW=FLOW+ARDEN*U(NIM1,J)
      UINC=(FLOWIN-FLOW)/ARDENT
      DO  215  J=2,NJM1
  215 U(NI,J)=U(NIM1,J)+UINC
      RETURN
C
CHAPTER  3  3  3  3  3  3  3  3  V MOMENTUM  3  3  3  3  3  3  3  3  3
C
    3 CONTINUE
C---------SIDE WALL
      IF(JSTEP .EQ. NJM1) GO TO 314
      CDTERM=CMU**0.25
      DO 325 I=2,NIM1
      IF(JMAX(I-1) .GE. JMAX(I)) GO TO 325
      JJ=JMAXP1(I-1)
      JI=JMAX(I)
      DO 320 J=JJ,JI
      XP=X(I)-XU(I)
      SQRTK=SQRT(0.5*(TE(I,J)+TE(I,J-1)))
      DENV=0.5*(DEN(I,J)+DEN(I,J-1))
      XPLUSA=0.5*(XPLUSW(J)+XPLUSW(J-1))
      IF(XPLUSA.LE.11.63)     GO  TO  311
      TMULT=DENV*CDTERM*SQRTK*CAPPA/ALOG(ELOG*XPLUSA)
      GO  TO  312
  311 TMULT=VISCOS/XP
  312 CONTINUE
  305 SP(I,J)=SP(I,J)-TMULT*SNSV(J)*RV(J)
      IF(J .EQ. JMAXP1(I-1)) SP(I,J)=SP(I,J)/2.
  310 AW(I,J)=0.0
  320 CONTINUE
  325 CONTINUE
C---------TOP WALL
  314 CONTINUE
      DO 313 I=2,NIM1
      J=JMAX(I)
  313 AN(I,J)=0.
      RETURN
C
CHAPTER  4  4  4  4  4  4  PRESSURE CORRECTION  4  4  4  4  4  4  4  4
C
    4 CONTINUE
C---------SIDE WALL
      IF(JSTEP .EQ. NJM1) GO TO 414
      DO 412 I=2,NIM1
      IF(JMAX(I-1) .GE. JMAX(I)) GO TO 412
```

```
      JJ=JMAXP1(I-1)
      JI=JMAX(I)
      DO 410 J=JJ,JI
      AW(I,J)=0.
  410 CONTINUE
  412 CONTINUE
C---------TOP WALL
  414 CONTINUE
      DO 402 I=2,NIM1
      J=JMAX(I)
  402 AN(I,J)=0.0
C---------SYMMETRY AXIS
      DO 420 I=2,NIM1
      AS(I,2)=0.0
  420 CONTINUE
C---------OUTLET
      DO 440 J=2,NJM1
      AE(NIM1,J)=0.0
  440 CONTINUE
      RETURN
C
CHAPTER  5  5  5  5  5  5  5  THERMAL ENERGY  5  5  5  5  5  5  5  5  5
C
    5 CONTINUE
C---------NO MODIFICATIONS FOR THIS PROBLEM
      RETURN
C
CHAPTER  6  6  6  6  6  TURBULENT KINETIC ENERGY  6  6  6  6  6  6  6  6
C
C
    6 CONTINUE
C---------TOP WALL
      CDTERM=CMU**0.25
      DO 610 I=2,NIM1
      J=JMAX(I)
      DWDY=(W(I,J+1)-W(I,J-1))/(DYNP(J)+DYPS(J))
      UAVG=U(I,J)*WFE(I)+(1.-WFE(I))*U(I+1,J)
      UEFF=SQRT(UAVG*UAVG + W(I,J)*W(I,J))
      YP=YV(J+1)-Y(J)
      DENU=DEN(I,J)
      SQRTK=SQRT(TE(I,J))
      VOL=RCV(J)*SNS(J)*SEW(I)
      YPLUSN(I)=DENU*SQRTK*CDTERM*YP/VISCOS
      IF(YPLUSN(I) .LE. 11.63) GO TO 608
      TMULT=DENU*CDTERM*SQRTK*CAPPA/ALOG(ELOG*YPLUSN(I))
      TAUN(I)=-TMULT*UEFF
      DITERM=DEN(I,J)*(CMU**.75)*SQRTK*ALOG(ELOG*YPLUSN(I))/(CAPPA*YP)
      GO TO 609
  608 TAURX=-VISCOS*UAVG/YP
      TAURW=VISCOS*(-W(I,J)/YP - W(I,J)/Y(J))
      TAUN(I)=SQRT(TAURX**2+TAURW**2)
      DITERM=DEN(I,J)*(CMU**.75)*SQRTK*YPLUSN(I)/YP
  609 DUDY=((U(I,J)+U(I+1,J)+U(I,J+1)+U(I+1,J+1))/4.-(U(I,J)+U(I+1,J)+U(
     4I,J-1)+U(I+1,J-1))/4.)/SNS(J)
      GENCOU=TAUN(I)**2/VIS(I,J)
      GENRES=GEN(I,J)-VIS(I,J)*(DUDY**2+(DWDY-W(I,J)/Y(J))**2)
      GEN(I,J)=GENRES+GENCOU
      SU(I,J)=GEN(I,J)*VOL+SUKD(I,J)
      SP(I,J)=-DITERM*VOL+SPKD(I,J)
      AN(I,J)=0.0
```

```
  610 CONTINUE
      TAUN(NI)=TAUN(NIM1)
C---------SIDE WALL
      IF(JSTEP .EQ. NJM1) GO TO 614
      DO 625 I=2,NIM1
      IF(JMAX(I-1) .GE. JMAX(I)) GO TO 625
      JJ=JMAXP1(I-1)
      JI=JMAX(I)
      DO 620 J=JJ,JI
      DWDX=(W(I+1,J)-W(I-1,J))/(DXPW(I)+DXEP(I))
      VAVG=V(I,J)*WFN(J)+(1.-WFN(J))*V(I,J+1)
      VEFF=SQRT(VAVG*VAVG + W(I,J)*W(I,J))
      XP=X(I)-XU(I)
      DENV=DEN(I,J)
      SQRTK=SQRT(TE(I,J))
      VOL=RCV(J)*SNS(J)*SEW(I)
      XPLUSW(J)=DENV*SQRTK*CDTERM*XP/VISCOS
      IF(XPLUSW(J) .LE. 11.63) GO TO 621
      TMULT=DENV*CDTERM*SQRTK*CAPPA/ALOG(ELOG*XPLUSW(J))
      TAUW(J)=-TMULT*VEFF
      DITERM=DEN(I,J)*(CMU**.75)*SQRTK*ALOG(ELOG*XPLUSW(J))/(CAPPA*XP)
      GO TO 622
  621 TAUXR=VISCOS*VAVG/XP
      TAUXW=VISCOS*W(I,J)/XP
      TAUW(J)=SQRT(TAUXR**2+TAUXW**2)
      DITERM=DEN(I,J)*(CMU**.75)*SQRTK*XPLUSW(J)/XP
  622 DVDX=((V(I,J)+V(I,J+1)+V(I+1,J)+V(I+1,J+1))/4.-(V(I,J)+V(I,J+1)+V(
     3I-1,J)+V(I-1,J+1))/4.)/SEW(I)
      GENCOU=TAUW(J)**2/VIS(I,J)
      GENRES=GEN(I,J)-VIS(I,J)*(DVDX**2+DWDX**2)
      GEN(I,J)=GENRES+GENCOU
      SU(I,J)=SU(I,J)+SUKD(I,J)+GEN(I,J)*VOL
      SP(I,J)=SP(I,J)+SPKD(I,J)-DITERM*VOL
      AW(I,J)=0.0
  620 CONTINUE
  625 CONTINUE
      TAUW(NJ)=TAUW(NJM1)
C---------SYMMETRY AXIS
  614 CONTINUE
      J=2
      DO 630 I=2,NIM1
      DUDY=((U(I,J)+U(I+1,J)+U(I,J+1)+U(I+1,J+1))/4.-(U(I,J)+U(I+1,J)+
     3U(I,J-1)+U(I+1,J-1))/4.)/SNS(J)
      VOL=RCV(J)*SNS(J)*SEW(I)
      GEN(I,J)=GEN(I,J)-VIS(I,J)*DUDY**2
      SU(I,J)=SUKD(I,J)+GEN(I,J)*VOL
  630 AS(I,2)=0.0
C---------OUTLET
      DO 640 J=2,NJM1
      AE(NIM1,J)=0.0
  640 CONTINUE
      RETURN
C
CHAPTER  7  7  7  7  7  7  7  7  DISSIPATION  7  7  7  7  7  7  7  7  7
C
    7 CONTINUE
C---------TOP WALL
      DO 710 I=2,NIM1
      J=JMAX(I)
      YP=YV(J+1)-Y(J)
```

```
      TERM=(CMU**.75)/(CAPPA*YP)
      SU(I,J)=GREAT*TERM*TE(I,J)**1.5
  710 SP(I,J)=-GREAT
C---------SIDE WALL
      IF(JSTEP .EQ. NJM1) GO TO 714
      DO 725 I=2,NIM1
      IF(JMAX(I-1) .GE. JMAX(I)) GO TO 725
      JJ=JMAXP1(I-1)
      JI=JMAX(I)
      DO 720 J=JJ,JI
      IF(J .EQ. JMAX(I)) GO TO 720
      XP=X(I)-XU(I)
      TERM=(CMU**.75)/(CAPPA*XP)
      SU(I,J)=GREAT*TERM*TE(I,J)**1.5
      SP(I,J)=-GREAT
  720 CONTINUE
  725 CONTINUE
C---------SYMMETRY  AXIS
  714 CONTINUE
      DO  730  I=2,NIM1
  730 AS(I,2)=0.0
C---------OUTLET
      DO 740 J=2,NJM1
      AE(NIM1,J)=0.0
  740 CONTINUE
      RETURN
C
CHAPTER 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
C     SWIRL VELOCITY
C
    8 CONTINUE
C---------TOP WALL
      CDTERM=CMU**0.25
      DO 810 I=2,NIM1
      J=JMAX(I)
      YP=YV(J+1)-Y(J)
      SQRTK=SQRT(TE(I,J))
      DENW=DEN(I,J)
      YPLUSA=YPLUSN(I)
      IF(YPLUSA .LE. 11.63) GO TO 811
      TMULT=DENW*CDTERM*SQRTK*CAPPA/ALOG(ELOG*YPLUSA)-VIS(I,J)/R(J)
      GO TO 812
  811 TMULT=VISCOS/YP
  812 SP(I,J)=SP(I,J)-TMULT*SEW(I)*RV(J+1)
      AN(I,J)=0.0
  810 CONTINUE
C---------SIDE WALL
      IF(JSTEP .EQ. NJM1) GO TO 814
      DO 855 I=2,NIM1
      IF(JMAX(I-1) .GE. JMAX(I)) GO TO 855
      JJ=JMAXP1(I-1)
      JI=JMAX(I)
      DO 850 J=JJ,JI
      XP=X(I)-XU(I)
      SQRTK=SQRT(TE(I,J))
      DENW=DEN(I,J)
      XPLUSA=XPLUSW(J)
      IF(XPLUSA .LE. 11.63) GO TO 851
      TMULT=DENW*CDTERM*SQRTK*CAPPA/ALOG(ELOG*XPLUSA)
      GO TO 852
```

```
851   TMULT=VISCOS/XP
  852 SP(I,J)=SP(I,J)-TMULT*SNS(J)*RCV(J)
      AW(I,J)=0.
  850 CONTINUE
  855 CONTINUE
C---------SYMMETRY AXIS
  814 CONTINUE
C---------FIX W FOR SOLID BODY ROTATION AT J=2 USING W AT J=3
      DO 860 I=2,NIM1
      TERM=W(I,3)*R(2)/R(3)
      SU(I,2)=GREAT*TERM
  860 SP(I,2)=-GREAT
C---------OUTLET
      DO 870  J=2,NJM1
870   AE(NIM1,J)=0.
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE STRMFN
CA*********************************************************************
C
CHAPTER  0  0  0  0  0  0  0  0  PRELIMINARIES  0  0  0  0  0  0  0  0
C
      COMMON
     1/VVEL/RESORV,NSWPV,URFV,DYNPV(24),DYPSV(24),SNSV(24)
     */VAR/U(48,24), V(48,24), W(48,24), P(48,24), PP(48,24), TE(48,24),
     *ED(48,24),STFN(48,24),YSTLN(48,24),STVAL(24),USTAR(48,24),
     *VSTAR(48,24),WSTAR(48,24),PSTAR(48,24),TESTAR(48,24),YSTLND(48,24)
     #,VISTAR(48,24)
     1/ALL/IT,JT,NI,NJ,NIM1,NJM1,GREAT,JMAX(48),JMAXP1(48)
     1/GEOM/INDCOS,X(48),Y(24),DXEP(48),DXPW(48),DYNP(24),DYPS(24),
     1      SNS(24),SEW(48),XU(48),YV(24),R(24),RV(24),
     #  WFN(24),WFS(24),WFE(48),WFW(48),RCV(24),XND(48),XUND(48),
     #YND(24),YVND(24)
     1/KASE T1/UIN,TEIN,EDIN,FLOWIN,ALAMDA,
     2       RSMALL,RLARGE,AL1,AL2,JSTEP,ISTEP,JSTP1,JSTM1,ISTP1,ISTM1
     1/PLOTT/NSTLN,NPLTLN,NPTS,YSLPLT(10,48),XUDPLT(48),INPLOT
      LOGICAL INPLOT
C
CHAPTER 1 1 1 1 1 CALCULATE STREAM FCN BASED ON VOLUMETRIC FLOW
C
C
      Q=UIN*(RSMALL**2)/2.
      DO 400 I=2,NI
      IF(JMAX(I-1) .LT. 5) GO TO 400
      STFN(I,2)=(Y(2)*R(2)*U(I,2)*.5)/Q
      JJ=JMAX(I-1)
      DO 200 J=3,JJ
      STFN(I,J)=STFN(I,J-1)+SNSV(J)*(R(J-1)*U(I,J-1)+R(J)*U(I,J))*.5/Q
  200 CONTINUE
  400 CONTINUE
C
      DO 800 I=2,NI
      IJ=JMAXP1(I-1)
      DO 700 K=1,NSTLN
      AK=K-1
      STVAL(K)=AK*.1
      JJ=JMAX(I-1)
```

```
      DO 600 J=2,JJ
      IF(STFN(I,J) .GE. STVAL(K)) GO TO 650
  600 CONTINUE
      YSTLN(I,K)=RV(IJ)
      GO TO 670
  650 IF(J .EQ. 2) YSTLN(I,K)=0.0
      IF(J .EQ. 2) GO TO 670
      SLOPE=(STVAL(K)-STFN(I,J-1))/(STFN(I,J)-STFN(I,J-1))
      YSTLN(I,K)=Y(J-1)+SLOPE*(Y(J)-Y(J-1))
  670 CONTINUE
      YSTLND(I,K)=YSTLN(I,K)/(2.*RLARGE)
  700 CONTINUE
  800 CONTINUE
      IF(.NOT. INPLOT) GO TO 745
      N=0
      DO 730 K=1,11,2
      N=N+1
      DO 730 I=1,NIM1
      YSLPLT(N,I)=YSTLND(I+1,K)
  730 CONTINUE
      DO 740 I=1,NIM1
      XUDPLT(I)=XUND(I+1)
  740 CONTINUE
  745 NPTS=NIM1
      RETURN
      END
C
C
      SUBROUTINE PLOT  (X,IDIM,IMAX,XAXIS,Y,JDIM,JMAX,YAXES,SYMB L,LA)
CA*****************************************************************
C
C  SUBROUTINE FOR PLOTTING J CURVES OF Y(J,I) AGAINST X(I).
C
C  X AND Y ARE ASSUMED TO BE IN ANY RANGE EXCEPT THAT NEGATIVE VALUES  *
C   ARE PLOTTED AS ZERO.                                                *
C  X AND Y ARE SCALED TO THE RANGE 0. TO 1. BY DIVISION BY THE MAXIMA, *
C   WHICH ARE ALSO PRINTED.
C  IDIM IS THE VARIABLE DIMENSION FOR X.
C  IMAX IS THE NUMBER OF X VALUES.
C  XAXIS STORES THE NAME OF THE X-AXIS.
C  JDIM IS THE VARIABLE DIMENSION FOR Y.
C  JMAX IS THE NUMBER OF CURVES TO BE PLOTTED, (UP TO 10).
C  THE ARRAY YAXES(J) STORES THE NAMES OF THE CURVES.
C  THE ARRAY SYMBOL(J) STORES THE SINGLE CHARACTERS USED FOR PLOTTING. *
C
CA****************************************************************
      DIMENSION X(IDIM),Y(JDIM,IDIM),YAXES(JDIM),SYMB L(JDIM),
     1 A(101),YMAX(10)
      DATA  DOT,CROSS,BLANK/1H.,1H+,1H /
C---------SCALING X ARRAY TO THE RANGE 0 TO 50
      XMAX=1.E-30
      DO 1 I=1,IMAX
       IF(X(I).GT.XMAX) XMAX=X(I)
    1 CONTINUE
      DO 2 I=1,IMAX
      X(I)=X(I)/XMAX*50.
       IF(X(I).LT.0.) X(I)=0.
    2 CONTINUE
C---------SCALING Y ARRAY TO THE RANGE 0 TO 100
      DO 3 J=1,JMAX
```

```
      YMAX(J)=1.E-30
      DO 4 I=1,IMAX
       IF(Y(J,I).GT.YMAX(J)) YMAX(J)=Y(J,I)
    4 CONTINUE
      DO 3 I=1,IMAX
C         NO Y  SCALING
      Y(J,I)=Y(J,I)*100.0
      IF(LA.EQ.1) Y(J,I)=0.02*Y(J,I)
       IF(Y(J,I).LT.0.) Y(J,I)=0.
    3 CONTINUE
C---------IDENTIFYING THE VARIOUS CURVES TO BE PLOTTED
      WRITE(6,103) XAXIS
      WRITE(6,100) (YAXES(I),I=1,JMAX)
      WRITE(6,106) (SYMB L(I),I=1,JMAX)
      WRITE(6,102) (YMAX(I),I=1,JMAX)
      DO 5 I=1,11
    5  A(I)=0.1*      (I-1)
      IF(LA .EQ. 1) WRITE(6,120)
      IF(LA .EQ. 0) WRITE(6,115)
      WRITE(6,101) (A(I),I=1,11)
C---------MAIN LOOP.  EACH PASS PRODUCES AN X-CONSTANT LINE.
      DO 40 II=1,51
      I=II
      IF(I.EQ.1.OR.I.EQ.51) GO TO 32
      GO TO 33
C---------ALLOCATE . OR + AS MARKER ON THE Y-AXIS
   32 DO 30 K=1,101
   30 A(K)=DOT
      DO 31 K=11,101,10
   31 A(K)=CROSS
C---------ALLOCATE . OR + MARK ON THE X-AXIS, ALSO THE APPROPRIATE
C---------      X VALUE
   33 A(1)=DOT
      A(101)=DOT
      K=I-1
   46 K=K-5
      IF(K)48,47,46
   47 A(1)=CROSS
      A(101)=CROSS
   48 XL=0.02*      (I-1)
C---------CHECK IF ANY Y( X(I) ) VALUE LIES ON THIS X-CONSTANT LINE
C---------IF YES GO TO 41, OTHERWISE GO TO 42
      DO 43 K=1,IMAX
      IFIX=X(K)+1.5
      IF(IFIX-I)43,41,43
C---------LOCATE  Y(  X(I)  )
   41 DO 44 J=1,JMAX
      NY=Y(J,K)+1.5
      A(NY)=SYMB L(J)
   44 CONTINUE
      GO TO 42
   43 CONTINUE
C---------PRINT X-CONSTANT LINE
  42  CONTINUE
      IF(LA.EQ.1) GO TO 51
      WRITE(6,105) XL,(A(K),K=1,101),XL
      GO TO 52
  51  WRITE(6,107) XL,(A(K),K=1,101),XL
  52  CONTINUE
C---------PUTTING BLANKS INTO X-CONSTANT LINE
```

```
      DO 49 K=1,101                                                        0198100
   49 A(K)=BLANK                                                           0198200
   40 CONTINUE                                                             0198300
      DO 50 I=1,11                                                         0198400
   50 A(I)=.1*      (I-1)                                                  0198500
      WRITE(6,104)(A(I),I=1,11)                                            0198600
      WRITE(6,130)                                                         0198700
      RETURN                                                               0198800
  100 FORMAT(11H Y-AXES ARE,5X,10(1X,A10))                                 0198900
  101 FORMAT(1H0,2X,11F10.1)                                               0199000
  102 FORMAT(15H MAXIMUM VALUES,  10E11.3)                                 0199100
  103 FORMAT(11H1X-AXIS IS ,A3)                                            0199200
  104 FORMAT(3X,11F10.1)                                                   0199300
  105 FORMAT(2H X,F6.2,3X,101A1,F6.2)                                      0199400
  106 FORMAT(7H SYMBOL,11X,10(1X,A10))                                     0199500
  107 FORMAT(/2H X,F6.2,3X,101A1,F6.2)                                     0199600
  115 FORMAT(//,T50,'RADIAL POSITION R/D',/)                               0199700
  120 FORMAT(//,T50,'RADIAL POSITION 2R/D',/)                              0199800
  130 FORMAT(///,T45,'DIMENSIONLESS STREAMLINE PLOT')                      0199900
      RETURN                                                               0200000
      END                                                                  0200100
                 U   VELOCITY                                              0200600
                 V   VELOCITY                                              0200700
                   W VELOCITY                                              0200800
                  PRESSURE                                                 0200900
               TEMPERATURE                                                 0201000
              TURBULENCE  ENERGY                                           0201100
            TURBULENCE  DISSIPATION                                        0201200
                   VISCOSITY                                               0201300
      DIMENSIONLESS LENGTH SCALE                                           0201400
      DIMENSIONLESS STREAM FUNCTION                                        0201500
    RADIAL COORDINATE OF STREAMLINES                                       0201600
     DIMENSIONLESS U VELOCITY                                              0201700
    DIMENSIONLESS V VELOCITY                                               0201800
    DIMENSIONLESS W VELOCITY                                               0201900
    DIMENSIONLESS PRESSURE                                                 0202000
    DIMENSIONLESS TURBULENCE ENERGY                                        0202100
    DIMENSIONLESS STREAMLINE COORDS                                        0202200
     DIMENSIONLESS EFF. VISCOSITY                                          0202300
```

1. Report No. NASA CR-3442	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle
A COMPUTER CODE FOR SWIRLING TURBULENT AXISYMMETRIC RECIRCULATING FLOWS IN PRACTICAL ISOTHERMAL COMBUSTOR GEOMETRIES

5. Report Date
February 1982

6. Performing Organization Code

7. Author(s)
D. G. Lilley and D. L. Rhode

8. Performing Organization Report No.
None

10. Work Unit No.

9. Performing Organization Name and Address
Oklahoma State University
School of Mechanical and Aerospace Engineering
Stillwater, Oklahoma 74078

11. Contract or Grant No.
NAG 3-74

13. Type of Report and Period Covered
Contractor Report

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington D. C. 20546

14. Sponsoring Agency Code
505-32-32

15. Supplementary Notes
Final report. Project Manager, Cecil J. Marek, Aerothermodynamics and Fuels Division, NASA Lewis Research Center, Cleveland, Ohio 44135. The microfiche supplement at the back of this report contains the output obtained for a sample computation.

16. Abstract
A primitive pressure-velocity variable finite difference computer code has been developed to predict swirling recirculating inert turbulent flows in axisymmetric combustors in general, and for application to a specific idealized combustion chamber with sudden or gradual expansion. The technique involves a staggered grid system for axial and radial velocities, a line relaxation procedure for efficient solution of the equations, a two-equation k-ϵ turbulence model, a stairstep boundary representation of the expansion flow, and realistic accommodation of swirl effects. This report is a user's manual and deals with the computational problem, showing how the mathematical basis and computational scheme may be translated into a computer program. A flow chart, Fortran 4 listing, notes about various subroutines and a user's guide are supplied as an aid to prospective users of the code.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Computer program Combustor Turbulent flow	Unclassified - unlimited STAR Category 07

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	133	A07

* For sale by the National Technical Information Service, Springfield, Virginia 22161